Marek Wydmuch

Addressing the long-tail problem in extreme multi-label classification

Doctoral Dissertation

Submitted to the Discipline Council
of Information and Communication
Technology
of Poznan University of Technology

Advisor: Krzysztof Dembczyński, Eng. Ph.D., Habil.

Poznan · 2025

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computing Science.

Marek Wydmuch
Machine Learning Laboratory
Faculty of Computing and Telecommunications
Institute of Computing Science
Poznan University of Technology
mwydmuch@cs.put.poznan.pl

The dissertation was typeset by the author in LaTeX.
The cover and back cover were designed by the author.

This dissertation and associated materials can be downloaded from:
https://www.cs.put.poznan.pl/mwydmuch/dissertation

Institute of Computing Science
Poznan University of Technology
Piotrowo 2, 60–965 Poznan, Poland
https://www.cs.put.poznan.pl

Computational experiments were partially performed in Poznan Supercomputing and Networking Center.

Abstract

Extreme multi-label classification (XMLC) is a learning task of assigning multiple labels to instances from an extremely large pool of possible labels, reaching an order of hundreds of thousands or even millions. Such scenarios are increasingly prevalent in modern applications like document and media tagging, recommendation systems, and web advertising. However, XMLC faces significant challenges, notably computational complexity and severe statistical issues arising from the sparsity of labels, commonly referred to as the "long-tail" problem. This dissertation specifically focuses on the "long-tail" labels, which, despite their infrequency, hold significant value in many real-world applications. However, the current XMLC metrics, such as precision@k, inadequately capture performance on these rare labels, allowing classifiers to ignore them without impacting the metric values, motivating the need for an alternative way of evaluating performance on the "long tail". Consequently, this thesis proposes the use of macro-averaged metrics, which average binary metrics calculated for each label, making them all equally important. The common settings in XMLC often require predicting exactly k labels. This constraint, combined with macro-averaged metrics, creates a unique optimization challenge that was not considered before. The core hypothesis examined in this dissertation asserts the existence of consistent inference algorithms for optimizing these macro-averaged metrics budgeted at k. The main contributions include deriving Bayes-optimal classifiers for both classic instance-wise and macro-averaged metrics. The latter is being analyzed under two distinct statistical frameworks – expected test utility (ETU) and population utility (PU). In all cases, regret bounds that establish consistency under realistic conditions are provided, and computationally efficient inference algorithms are proposed. Empirical evaluations validate theoretical results, demonstrating that introduced methods significantly improve results on macro-averaged metrics. At the same time, they can maintain good performance on classical metrics and are computationally efficient, making them suitable for the XMLC problems.

Acknowledgments

I wish to express my sincere gratitude to everyone who has taught me something throughout my life.

In particular, I would like to extend my deepest thanks to my advisor, Krzysztof Dembczyński, for the opportunity to experience both the joys and hardships of doing research, as well as for his unlimited guidance, kindness, and support.

I am also deeply grateful to Wojciech Jaśkowski, who sparked my interest in machine learning and guided my first steps in doing research.

My sincere appreciation goes to all my co-authors. I greatly enjoyed working with all of you. I would especially like to thank Kalina Jasinska-Kobus, Erik Schultheis, Rohit Babbar, and Wojciech Kotłowski, who have collaborated with me on much of this work.

I am thankful to Prof. Roman Słowiński, Prof. Jerzy Stefanowski, and all the members of the Intelligent Decision Support Systems Laboratory and Machine Learning Laboratory at Poznan University of Technology for creating a kind and supportive atmosphere during my studies.

Last but certainly not least, I wish to thank my friends and family, without whom I might have finished this dissertation much earlier or perhaps never at all, as I am not sure it would have been worth finishing without them. I would like to thank my folks from Rataje, Lyceum No. 8, Gutkowo, ML in PL Association, and my fellow PhD students for their camaraderie and many good laughs.

From the bottom of my heart, I thank my parents, Anna and Jacek, for their unconditional love and for nurturing my curiosity, my sister Marta for always caring about me, my dearest love, Magdalena, for her never-ending support, and our daughters, Matylda, Marianna, and Lilianna, for perhaps the most important life lesson.

List of publications

Below, we list all scientific publications co-authored by Marek Wydmuch related to this dissertation:

W. Kotłowski, M. Wydmuch, E. Schultheis, R. Babbar, and K. Dembczyński. A General Online Algorithm for Optimizing Complex Performance Metrics. In Forty-first International Conference on Machine Learning, 2024,

E. Schultheis, W. Kotłowski, M. Wydmuch, R. Babbar, S. Borman, and K. Dembczyński. Consistent algorithms for multi-label classification with macro-at-k metrics. In The Twelfth International Conference on Learning Representations, 2024,

E. Schultheis, M. Wydmuch, W. Kotłowski, R. Babbar, and K. Dembczyński. Generalized test utilities for long-tail performance in extreme multi-label classification. In Advances in Neural Information Processing Systems, volume 36, pages 22269–22303. Curran Associates, Inc., 2023,

E. Schultheis, M. Wydmuch, R. Babbar, and K. Dembczyński. On Missing Labels, Long-tails and Propensities in Extreme Multi-label Classification. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22, page 1547–1557, New York, NY, USA, 2022. Association for Computing Machinery,

M. Wydmuch, K. Jasinska-Kobus, R. Babbar, and K. Dembczyński. Propensity-Scored Probabilistic Label Trees. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21, page 2252–2256, New York, NY, USA, 2021. Association for Computing Machinery,

K. Jasinska-Kobus, M. Wydmuch, D. Thiruvenkatachari, and K. Dembczyński. Online probabilistic label trees. In Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, volume 130 of Proceedings of Machine Learning Research, pages 1801–1809. PMLR, 13–15 Apr. 2021,

K. Jasinska-Kobus, M. Wydmuch, K. Dembczynski, M. Kuznetsov, and R. Busa-Fekete. Probabilistic Label Trees for Extreme Multi-label Classification. 2020,

M. Wydmuch, K. Jasinska, M. Kuznetsov, R. Busa-Fekete, and K. Dembczyński. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In Advances in Neural Information Processing Systems, volume 31, pages 6358–6368. Curran Associates, Inc., 2018.

List of changes

Below is a list of changes in this version compared to the submitted version of the dissertation.

- Removed the Polish title page, abstract, introduction, and synopsis as they are redundant and not relevant for English speaking readers.
- The list of scientific publications co-authored by the author of this dissertation was edited to contain only publications relevant to this dissertation.
- Fixed typos, missing or excessive commas, and grammar throughout the dissertation without changing the intended meaning.
- Improved the sizes of parentheses, line breaks, and function-application notation in equations and figures throughout the dissertation.
- Fixed the definition and notation of the multi-label confusion matrix, including the missing transpose in its matrix representation and related uses of $\widehat{\mathbf{C}}$ in the definitions of utilities based on confusion matrices (Chapter 2).
- Corrected several mathematical and notational issues: the false-positive entry of the binary confusion matrix, the notation for the PU expected utility/risk, and the definition of the inter-label imbalance ratio (Chapter 2); the alternative parameterization of Jaccard similarity (Chapter 5); the rate in the ETU approximation bound and the subscript in Ψ_{Cov} and ψ_{Cov} (Chapter 6); and the empirical true-positive notation in the PU appendix (Appendix A.4).
- Updated the sparse-inference complexity discussion, including the Frank-Wolfe step complexity and the dependence on the retained top-k' labels (Chapter 8).
- Improved notation consistency and clarified the probability/regret notation used in the PLT analysis (Chapter 8).
- Updated bibliography entries by removing duplicates, and preserving original capitalization in titles.

Contents

Notation

General conventions of the notation and nomenclature

Before starting the main content of this dissertation, we introduce some conventions that we use throughout the text. We use:

- italic lowercase symbols to denote scalars, e.g., a, italic bold lowercase symbols to denote vectors, e.g., $\boldsymbol{a}$, and upright bold uppercase symbols to denote matrices, e.g., $\mathbf{A}$,
- a_i to denote the i-th element of vector $\boldsymbol{a}$, $\boldsymbol{a}_i$ to denote i-th row vector of matrix $\mathbf{A}$, $\boldsymbol{a}_{:,j}$ to denote j-th column vector of matrix $\mathbf{A}$, and $a_{i,j}$ to denote a single element of matrix $\mathbf{A}$ (i-th row and j-th column),
- calligraphic symbols to denote sets, e.g., $\mathcal{A}$,
- $[\cdot]$ to define a vector, e.g., $\boldsymbol{a} = [a_1, \ldots, a_n]$, a vector of vectors is sometimes used to define a matrix, e.g., $\mathbf{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n]$ where $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ are rows, and $\mathbf{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n]^\intercal$ where $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ are columns of $\mathbf{A}$,
- $\{\cdot\}$ to define a set, e.g., $\mathcal{A} = \{a_1, \ldots, a_n\}$,
- $(\cdot)$ to define a tuple (collection of elements of different types), e.g., (a, b),
- $[i] := \{1, 2, \ldots, i\}$ to denote a set of first i natural numbers.
- $\|\cdot\|_p$ to denote L_p-norm of a vector or matrix, and $|\cdot|$ the size (number of elements) of a vector, matrix, set, or list,
- $\mathbb{1}[\cdot]$ to denote an indicator function, e.g., $\mathbb{1}[a > 0] = 1$ if $a > 0$ else 0,
- $\boldsymbol{a} \cdot \boldsymbol{b}$ to denote dot product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ and $\mathbf{A} \cdot \mathbf{B}$ to denote a matrix dot product of $\mathbf{A}$ and $\mathbf{B}$,
- $\boldsymbol{a} \odot \boldsymbol{b} = [a_1 b_1, a_2 b_2, \ldots, a_n b_n]$ to denote the element-wise (Hadamard) product (multiplication) of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ of size n,
- blue and orange colors are sometimes used to highlight some parts of equations to make them easier to read.

Below we present a list of the most commonly used symbols with a short explanation. All the presented symbols are also reintroduced in the body of the dissertation for the reader's convenience.

Instances and labels

n	Number of samples
d	Number of features
$\mathcal{X}$	Instance space
$\boldsymbol{x}$	Single instance $\boldsymbol{x} \in \mathcal{X}$
$\mathbf{X}$	Matrix of instances $\mathbf{X} := [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]$
m	Number of labels
$\mathcal{Y}$	Space of labels vectors, $\mathcal{Y} := \{0, 1\}^m$
y	Single binary label, $y \in \{0, 1\}$, $y = 1$ means the label is relevant (positive), and $y = 0$ means the label is irrelevant (negative)
$\boldsymbol{y}$	Labels vector, $\boldsymbol{y} \in \mathcal{Y}$, $\boldsymbol{y} := [y_1, \cdots, y_m]$
$\mathbf{Y}$	Matrix of labels, where each row is labels vector, $\mathbf{Y} := [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n]$
$\mathcal{L}$	Set of relevant labels, $\mathcal{L} \subseteq [m]$
$\mathcal{D}$	Dataset, a list of tuples with instances and labels, $\mathcal{D} := [(\boldsymbol{x}_i, \boldsymbol{y}_i)]_{i=1}^n$
π_j	Prior probability of a single label j being relevant, $\pi_j := \mathbb{P}(y_j = 1)$
$\boldsymbol{\pi}$	Vector of prior probabilities of labels, $\boldsymbol{\pi} := [\pi_1, \cdots, \pi_m]$
$\eta_j(\boldsymbol{x})$	Marginal conditional probability of a single label j being relevant for instance $\boldsymbol{x}$, $\eta_j(\boldsymbol{x}) := \mathbb{P}(y_j = 1 \mid \boldsymbol{x})$
$\boldsymbol{\eta}(\boldsymbol{x})$	Vector of marginal conditional probabilities of labels, $\boldsymbol{\eta}(\boldsymbol{x}) := [\eta_1(\boldsymbol{x}), \cdots, \eta_m(\boldsymbol{x})]$

Classifiers and predictions

$\mathcal{H}$	Classifier hypothesis space
$\boldsymbol{h}$	Classifier
$\widehat{y}$	Single binary predicted label, $\widehat{y} \in \{0, 1\}$
$\widehat{\boldsymbol{y}}$	Vector of predicted labels, $\widehat{\boldsymbol{y}} \in \mathcal{Y}$, $\widehat{\boldsymbol{y}} := [\widehat{y}_1, \cdots, \widehat{y}_m]$
$\widehat{\mathbf{Y}}$	Matrix of predicted labels, $\widehat{\mathbf{Y}} := [\widehat{\boldsymbol{y}}_1, \cdots, \widehat{\boldsymbol{y}}_n]$
$\widehat{\eta}_j(\boldsymbol{x})$	Estimate of $\eta_j(\boldsymbol{x})$ coming from Label Probability Estimator (LPE)
$\widehat{\boldsymbol{\eta}}(\boldsymbol{x})$	Estimate of $\boldsymbol{\eta}(\boldsymbol{x})$, $\widehat{\boldsymbol{\eta}}(\boldsymbol{x}) := [\widehat{\eta}_1(\boldsymbol{x}), \cdots, \widehat{\eta}_m(\boldsymbol{x})]$

$\boldsymbol{h}^{@k}$	Classifier budgeted at k
$\mathcal{Y}^{@k}$	Space of prediction vectors with exactly k positives $\mathcal{Y}^{@k} := \{0,1\}^m$,
$\widehat{\boldsymbol{y}}^{@k}$	Vector of predicted labels with k positives $\widehat{\boldsymbol{y}}^{@k} \in \mathcal{Y}^{@k}$
$\widehat{\mathbf{Y}}^{@k}$	Matrix of predicted labels with k positives for each row
θ	Sampling probability of a label j in a randomized classifier
$\boldsymbol{\theta}$	Vector of sampling probabilities
$\boldsymbol{h}^{\text{rnd}@k}$	Randomized classifier budgeted at k
$\Delta^{@k}$	Sampling probability space
$\mathcal{S}^{@k}$	Set of all sampling estimation functions

Confusion matrices and utilities

$\mathcal{C}$	Space of binary confusion matrices
$\mathcal{C}^m$	Space of multi-label confusion matrices
$\mathcal{C}^{m,@k}$	Space of multi-label confusion matrices calculated for prediction budgeted at k
$\widehat{c}$	Single entry of a confusion matrix
$\widehat{\boldsymbol{c}}$	Binary confusion matrix (vector of true negatives, false positives, false negatives, true positives)
$\widehat{\mathbf{C}}$	Multi-label confusion matrix
c	Expected value of confusion matrix entry
$\boldsymbol{c}$	Expected binary confusion matrix (vector)
$\mathbf{C}$	Expected multi-label confusion matrix
Ψ	Multi-label Utility
ψ	Binary utility
$\Phi(\boldsymbol{h})$	Expected utility/Risk of the classifier $\boldsymbol{h}$
$\text{Reg}(\boldsymbol{h})$	Regret of the classifier $\boldsymbol{h}$

Missing labels setting

$\breve{y}$	Single observed binary label
$\breve{\boldsymbol{y}}$	Observed labels vector
$\breve{\mathbf{Y}}$	Observed matrix of labels
p	Propensity of a label to be observed as positive

$\mathbf{P}$ Matrix of propensities

$\breve{\pi}_j$ Observed prior probability of a single label

$\breve{\boldsymbol{\pi}}$ Observed vector of label priors

$\breve{\eta}_j(\boldsymbol{x})$ Observed marginal probability of a single label

$\breve{\boldsymbol{\eta}}(\boldsymbol{x})$ Vector of observed marginal probabilities of labels

$\breve{\mathbf{C}}$ Observed confusion matrix

Probabilistic label trees

$\mathcal{T}$ Label tree

$\mathcal{V}$ Set of tree nodes in $\mathcal{T}$

v Single node $v \in \mathcal{V}$

v_{root} Root node

$\text{pa}(v)$ Node that is a parent of node v in $\mathcal{T}$

$\text{Ch}(v)$ Set of nodes that are children of node v in $\mathcal{T}$

$\text{lb}(v)$ Label assigned to node v

$\text{v}(j)$ Node that is assigned to label j

$\text{Labels}(v)$ Set of labels assigned to nodes in a subtree of node v

$\text{Path}(v)$ Set of nodes on the path from root node v_{root} to node v

$\eta_v^{\mathcal{T}}(\boldsymbol{x})$ Marginal conditional probability of a node v to contain a relevant label

$\eta_v^{\mathcal{T}}(\boldsymbol{x})$ Estimate of marginal conditional probability of a node v to contain a relevant label

$\mathcal{Q}$ First-in-first-out queue

$\mathcal{Q}^{\text{P}}$ Priority queue

1 Introduction

1.1 Machine learning and extreme multi-label classification

Machine learning is currently a broad and fast-growing sub-field of artificial intelligence that sits between the domains of computer science, mathematical optimization, information theory, statistics, and even neuroscience. The fundamental idea behind machine learning is to enable computers to discover algorithms for solving specific tasks that would be too difficult or too costly to be solved by human programmers. Instead of being provided with explicit instructions, a computer is presented with data related to the problem (e.g., examples of correctly performed tasks) and "learns" how to solve it based on that data. Recent advancements in the development of machine learning have led to its being successfully applied to a growing number of applications, which leads to the emergence of new research directions. At a high level, machine learning approaches are traditionally divided into three categories, which correspond to learning paradigms depending on the nature of the data available to the learning system:

- Supervised learning: A computer is presented with example inputs and their desired outputs (labels), given by a "teacher". We call the set of such examples a training set. The goal is to learn from these a general function that maps inputs to outputs.
- Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find hidden structure in its inputs.
- Reinforcement learning: A computer program (usually called an agent in this context) interacts with a dynamic environment in which it must achieve a certain goal. As it performs different actions, the environment provides feedback that is analogous to rewards. The goal is to learn a strategy that maximizes the sum of the rewards.

This categorization is a rather simplistic attempt at pigeonholing numerous machine

learning settings and algorithms.

Nevertheless, in this thesis, we will focus directly on the sub-class of supervised learning called classification. Classification involves assigning a discrete label, or a number of them, to an input instance based on its features as correctly as possible (e.g., naming objects visible in the image, diagnosing a disease based on patient results, assigning relevant categories and tags to the article, and predicting products that are likely to be bought by a specific customer). Classification is one of the most widely studied and applied machine learning tasks across numerous domains.

Simple classification tasks typically involve assigning a single label to each instance (multi-class classification) or determining the presence or absence of a single characteristic (binary classification). However, many real-world problems require assigning multiple labels to a single instance simultaneously. This type of classification problem, known as multi-label classification, introduces additional complexity compared to binary or multi-class classification. For example, a news article might belong to multiple categories such as "politics," "economy," and "international affairs"; an image might contain multiple objects like "person," "car," and "building"; a customer might buy many products from the store; or a patient might have multiple medical conditions simultaneously.

Extreme multi-label classification (XMLC) extends the multi-label classification paradigm to settings with extraordinarily large numbers of potential labels, ranging from thousands to millions, with only a small subset of a few being relevant for each input instance. This scenario is increasingly common in modern applications such as:

1. Documents and media tagging for search: Identifying relevant topics, categories, or keywords in documents and media (text, image, sound, video) from a vocabulary of millions of possible tags [Dekel and Shamir, 2010, Deng et al., 2011, Agrawal et al., 2013].
2. Recommendation: Suggesting a small subset of relevant items from a huge inventory to customers or matching them with other items [Weston et al., 2013, Zhuo et al., 2020, Song et al., 2020], or search queries [Medini et al., 2019, Chang et al., 2020].
3. Web advertising: Matching ads to search queries or web pages from a large collection of advertisements [Prabhu and Varma, 2014, Jain et al., 2019].

The scale of these problems makes XMLC different from standard multi-label classification, requiring different techniques and algorithms. In XMLC, both the feature space and labels are high-dimensional, posing several key challenges:

1. Computational efficiency: multi-label classification methods often have linear complexity with the number of labels or higher to account for complex dependencies between labels. Applying these methods becomes impractical when dealing with a large number of labels. Efficient algorithms that can train and predict in sublinear time are essential in this setting.
2. Statistical challenges: with a large number of potential labels, most of them

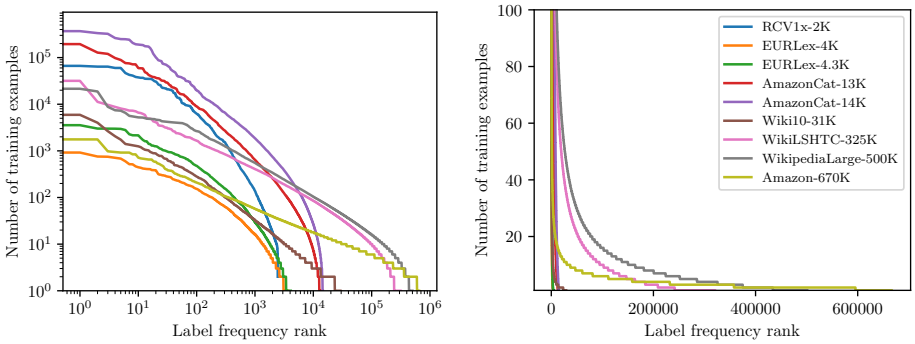


Figure 1.1: Label frequency in XMLC datasets. The x-axis shows the label rank when sorted by the frequency of positive instances and the y-axis gives the number of the positive instances. The first plot uses a log scale for both the x- and y-axis; the second uses a linear scale but crops the y-axis at 100.

appear in only a tiny fraction of instances, leading to severe data sparsity and imbalance issues. A large pool of labels also makes labeling prone to noise, especially labels going missing, as it is nearly impossible for annotators to carefully evaluate the entire set of labels when it is so large.

1.2 Long-tailed label distribution

While in this thesis, we touch on the aspect of efficient algorithms, we especially focus on the issue of high label imbalance and sparsity. Given the large number of labels in XMLC datasets reaching hundreds of thousands or even millions, and their adherence to Zipf's law [Adamic and Huberman, 2002], it is not surprising that many labels are very sparse, and therefore the label distribution is strongly "long-tailed" [Bhatia et al., 2015, Babbar and Schölkopf, 2017]. In Figure 1.1, we plot the distribution of label frequencies of nine popular benchmark datasets[1] from the XMLC repository [Bhatia et al., 2016], clearly showing that only a small fraction of labels are well represented in the whole set, while the rest form a long tail with very few examples each.

Paradoxically, in many applications, these rare labels are considered to be more important, as being more informative or more satisfying when predicted correctly, e.g., very general tags are rarely useful for finding information, the user feels better when a good recommendation of an item that was unknown to them instead of most popular one. Posing the critical research question, what is the best way to accommodate these tail labels.

[1] We will use the same set of nine datasets throughout this thesis.

1.3 Performance metrics in XMLC

When training a classifier, one usually aims to achieve the highest quality of predictions, expressed in the form of a metric that scores the classifier's output in comparison to the expected output. To evaluate the classifier's generalization capabilities, the additional (test) set, which has not been seen by the classifier before, is used. The test set, like the training set, needs to contain both the input and the expected output pairs. The simplest metric in classification is zero-one accuracy, which assigns a score of 1 if the classifier's prediction exactly matches the expected output for a given input and 0 otherwise. The scores are then averaged over all instances in the test set.

Due to the specificity of the setting, the XMLC area adopted its own set of standard metrics for evaluating classifiers. In traditional multi-label classification, the classifier's decision is evaluated either for each label or at the level of the whole positive and negative label sets. Due to the nature of XMLC applications, like search and recommendation, one usually does not care about the precise decision for each label, but rather obtaining a set of labels, often of specific size, that will maximize the evaluation metric. The size of the prediction set can often be indicated by a graphical interface having a specific number of slots for presenting search results, recommendations, or ads [Cremonesi et al., 2010, Chang et al., 2021]. To reflect that, it is common to evaluate using the so-called "at k" (@k) metrics, for which the classifier returns precisely k labels for an instance (sometimes also with a specified order). The most popular metric is precision@k, which counts how many of the k predicted labels were in the set of true labels, followed by metrics like recall@k, or (n)DCG@k.

It has been noticed that, under high label imbalance, competitive performance on these metrics can be achieved by correctly predicting the most popular labels [Jain et al., 2016, Wei and Li, 2020]. Therefore, the community has started to develop new metrics that prefer "rewarding" [Ye et al., 2020], "diverse" [Babbar and Schölkopf, 2019], and "rare and informative" [Prabhu et al., 2018a] labels over the frequently occurring head labels.

Jain et al. [2016] addressed this problem as the first ones by introducing propensity-scored performance metrics, which quickly gained popularity and became a standard in the XMLC community. These metrics are weighted variants of standard metrics like precision@k, that give increased weight to tail labels, but these weights have been derived from the perspective of missing labels, and the interpretation of the results on these metrics is not straightforward.

1.4 Motivation

We have noticed that even the popular propensity-scored precision only slightly better distinguishes classifiers that are good at predicting tail labels from the ones

that are not. This observation motivates the search for alternative metrics better suited to evaluating performance in long-tail scenarios.

In multi-label classification, one can consider a wide spectrum of measures that are usually divided into three categories based on the averaging scheme, namely instance-wise, micro-, and macro-averaging. Instance-wise measures are defined, as the name suggests, on the level of a single instance. Typical examples are already mentioned precision@k, recall@k, (n)DCG@k, Hamming loss, and the instance-wise F_β-measure. Micro-averages are defined on a single confusion matrix that accumulates true positives, false positives, false negatives, and true negatives from all the labels. Macro-averages require a binary metric to be applied to each label separately and then averaged over the labels. In general, any binary metric can be applied in any of the above averaging schemes. Not surprisingly, some of the metrics, for example, Hamming loss, lead to the same form of the final metric regardless of the scheme used. One can also consider the wider class of measures that are defined as general aggregation functions of label-wise confusion matrices.

The macro-averaged metric seems to be very attractive in the context of evaluating long-tail performance in XMLC, as they treat all the labels equally important, preventing the labels with a small number of positive examples from being overshadowed. Because of that, we take a closer look at these metrics with a prediction budget of k, which naturally fits the XMLC setting. The budget requires the prediction algorithm to choose labels "wisely," often leading to the presence of long-tail labels in the set of predicted labels. It is also natural in typical applications of XMLC, like recommendation systems, in which the number of slots is dictated by the user interface.

To illustrate the emphasis that macro-averaged put on tail-labels, in Table 1.1, we compare two classifiers,[2] The first was trained on all labels, and the second was trained only on 20% of the head labels, with the rest being discarded as they would not exist. Then, both models are evaluated using a complete set of labels. The standard metrics are only slightly perturbed by reducing the label space to the head labels. The propensity-scored precision decreases more significantly, sometimes even to 35%, but this drop in performance does not fully reflect the inability of a classifier to predict 80% of labels. In contrast, macro-averaged metrics (precision, recall, and F_1-measure) with a budget at k, as well as coverage@k, a popular auxiliary measure in XMLC [Jain et al., 2016, Babbar and Schölkopf, 2019, Wei and Li, 2020, Schultheis et al., 2022], decrease much more drastically, even over 70%, much better reflecting the classifier's inability to predict the remaining 80% of labels. These results show that budgeted-at-k macro measures might be very attractive in the context of long tails.

[2]We use an ensemble of probabilistic label trees [Jasinska-Kobus et al., 2020], the same method we use later in Chapter 9 for other experiments.

Table 1.1: Results (%) of a classifier trained on the full set of labels and a classifier trained with only 20% of head labels (most frequent labels) on different metrics budgeted at k (@k). The difference smaller than 5% is colored green, smaller than 30% orange, and bigger using red.

Metric	With all labels			With top 20% labels					
	@1	@3	@5	@1	(diff.)	@3	(diff.)	@5	(diff.)
RCV1x-2K									
Precision	89.99	72.18	51.67	89.96	(-0.03%)	72.08	(-0.14%)	51.57	(-0.21%)
Propensity-scored Prec.	96.94	78.69	56.92	96.84	(-0.11%)	78.36	(-0.42%)	56.52	(-0.72%)
Recall	40.31	74.60	81.17	40.29	(-0.06%)	74.49	(-0.14%)	81.02	(-0.19%)
nDCG	89.99	75.87	60.65	89.96	(-0.03%)	75.79	(-0.11%)	60.56	(-0.15%)
Macro-Precision	10.04	13.79	13.37	8.93	(-11.05%)	9.26	(-32.81%)	7.37	(-44.86%)
Macro-Recall	1.24	4.65	7.61	1.16	(-6.32%)	3.81	(-18.16%)	5.66	(-25.68%)
Macro-F_1	1.74	5.44	7.62	1.61	(-7.45%)	4.28	(-21.38%)	5.34	(-29.89%)
Coverage	12.42	26.14	35.91	11.24	(-9.51%)	17.18	(-34.27%)	18.89	(-47.39%)
EURLex-4.3K									
Precision	91.40	81.21	68.50	90.55	(-0.93%)	79.30	(-2.35%)	65.88	(-3.83%)
Propensity-scored Prec.	151.59	139.26	120.68	144.54	(-4.65%)	127.92	(-8.14%)	107.07	(-11.28%)
Recall	20.81	53.30	71.61	20.59	(-1.07%)	51.88	(-2.66%)	68.68	(-4.09%)
nDCG	91.40	83.62	74.24	90.55	(-0.93%)	81.95	(-2.00%)	72.01	(-3.01%)
Macro-Precision	14.39	22.55	25.36	11.84	(-17.70%)	13.83	(-38.69%)	12.22	(-51.83%)
Macro-Recall	4.45	14.15	22.57	3.26	(-26.71%)	9.04	(-36.10%)	12.97	(-42.53%)
Macro-F_1	6.16	16.41	22.73	4.69	(-23.99%)	10.40	(-36.61%)	12.20	(-46.33%)
Coverage	15.62	27.09	34.54	13.18	(-15.59%)	18.31	(-32.41%)	19.46	(-43.66%)
AmazonCat-14K									
Precision	89.28	69.10	54.63	89.25	(-0.03%)	68.77	(-0.47%)	52.99	(-3.00%)
Propensity-scored Prec.	95.23	77.83	64.44	94.32	(-0.96%)	75.77	(-2.65%)	59.86	(-7.10%)
Recall	39.40	69.11	83.63	39.40	(-0.01%)	68.84	(-0.39%)	81.73	(-2.26%)
nDCG	89.28	73.43	62.34	89.25	(-0.03%)	73.18	(-0.34%)	61.14	(-1.91%)
Macro-Precision	25.19	39.74	41.82	12.81	(-49.15%)	13.41	(-66.26%)	10.94	(-73.83%)
Macro-Recall	2.74	15.50	36.67	1.32	(-51.87%)	6.83	(-55.91%)	12.73	(-65.27%)
Macro-F_1	4.53	20.09	36.57	2.18	(-51.97%)	8.29	(-58.75%)	11.39	(-68.87%)
Coverage	30.25	54.89	69.59	15.57	(-48.52%)	18.91	(-65.56%)	19.87	(-71.45%)
WikiLSHTC-325K									
Precision	63.44	41.94	31.14	59.32	(-6.49%)	37.75	(-9.98%)	27.57	(-11.45%)
Propensity-scored Prec.	163.80	118.44	90.78	133.40	(-18.56%)	89.06	(-24.80%)	66.02	(-27.27%)
Recall	28.11	47.00	54.58	25.39	(-9.68%)	40.54	(-13.74%)	46.18	(-15.40%)
nDCG	63.44	46.72	37.94	59.32	(-6.49%)	42.51	(-9.00%)	34.16	(-9.95%)
Macro-Precision	12.83	18.61	18.73	7.10	(-44.64%)	7.25	(-61.06%)	6.03	(-67.82%)
Macro-Recall	6.53	15.71	20.72	3.57	(-45.42%)	7.38	(-53.00%)	9.17	(-55.75%)
Macro-F_1	7.78	15.19	17.30	4.16	(-46.61%)	6.33	(-58.34%)	6.31	(-63.54%)
Coverage	16.61	29.71	35.53	10.80	(-34.97%)	15.64	(-47.38%)	17.10	(-51.87%)
WikipediaLarge-500K									
Precision	66.83	47.79	37.41	60.88	(-8.91%)	42.00	(-12.11%)	32.46	(-13.22%)
Propensity-scored Prec.	187.95	142.69	113.72	139.25	(-25.91%)	98.25	(-31.15%)	76.40	(-32.82%)
Recall	21.86	39.77	48.07	18.75	(-14.20%)	32.43	(-18.45%)	38.55	(-19.81%)
nDCG	66.83	52.07	43.72	60.88	(-8.91%)	46.20	(-11.26%)	38.45	(-12.05%)
Macro-Precision	13.09	19.92	21.04	5.98	(-54.31%)	6.39	(-67.93%)	5.74	(-72.71%)
Macro-Recall	6.64	16.16	21.83	2.77	(-58.36%)	5.81	(-64.08%)	7.47	(-65.80%)
Macro-F_1	7.84	15.94	18.99	3.23	(-58.76%)	5.15	(-67.67%)	5.51	(-70.99%)
Coverage	16.58	30.83	37.87	9.31	(-43.89%)	13.92	(-54.85%)	15.66	(-58.64%)
Amazon-670K									
Precision	44.97	40.20	36.71	41.01	(-8.80%)	34.38	(-14.48%)	28.95	(-21.13%)
Propensity-scored Prec.	288.28	273.84	259.09	229.95	(-20.23%)	197.65	(-27.82%)	167.45	(-35.37%)
Recall	9.35	23.31	34.39	8.00	(-14.47%)	19.05	(-18.27%)	26.19	(-23.83%)
nDCG	44.97	41.29	38.58	41.01	(-8.80%)	35.91	(-13.03%)	31.75	(-17.70%)
Macro-Precision	4.78	10.46	14.21	3.38	(-29.24%)	5.31	(-49.20%)	5.43	(-61.77%)
Macro-Recall	3.08	9.04	14.41	2.04	(-33.75%)	5.20	(-42.51%)	7.23	(-49.82%)
Macro-F_1	3.42	9.01	13.39	2.26	(-33.74%)	4.70	(-47.82%)	5.59	(-58.23%)
Coverage	5.82	13.78	19.79	4.64	(-20.33%)	9.02	(-34.51%)	11.01	(-44.37%)

In this thesis, we examine the long-tail label problem through the lens of evaluation metrics. We propose the usage of macro-averaged metrics budgeted at k as an alternative, focusing on "tail-labels" performance and being easy to interpret and investigate the problem of optimizing them. Interestingly, optimization of the macro-averaged metrics can be considered under two distinct statistical frameworks – expected test utility (ETU) (also known in the literature as the decision theoretic approach (DTA) and population utility (PU) (also known as empirical utility maximization (EUM) [Ye et al., 2012, Dembczyński et al., 2017]. Under the ETU framework, the goal is to optimize the expected value of a metric on a given test set. While PU aims to optimize a metric value calculated on the expected value of a confusion matrix over the population. While the macro-averaged metrics, as well as a more general class of metrics defined on the label-wise confusion matrices, have been well-studied in multi-label classification under both frameworks, the budgeted @k variants have not, to the best of our knowledge. Furthermore, the previously introduced optimization frameworks cannot be directly applied in the budgeted setup. Since the optimization problems for different labels are tightly coupled with each other through the budget of k predictions, the final problem is much more difficult.

One of the first results of this work is an observation that there exists a class of label-wise metrics budgeted at k, that can be linearly decomposed over labels into binary utilities, which includes both standard instance-wise weighted metrics, like precision@k, and macro-averaged metrics, possibly allowing to obtain general results under a unified framework. Because of that, we are interested in investigating whether consistent inference algorithms exist to optimize label-wise metrics budgeted at k and if they are compatible with existing classifiers for XMLC. Due to extremely large label space, almost all existing XMLC methods aim to provide top k candidates with scores assigned to individual labels. Preferably, these scores reflect (or can be calibrated to reflect) estimates of the label marginal conditional probabilities. To be practical, an algorithm for optimizing label-wise metrics budgeted at k should operate over the marginal conditional probabilities of labels, allowing it to work on top of existing and future methods by being agnostic to their architecture. Additionally, we would like such an algorithm to be consistent, meaning that with the growing size of the training set, the regret of the classifier decreases to zero.

1.5 Aim and scope

Given the motivations presented above, the hypothesis of this dissertation is formulated as follows:

> **There exist consistent inference algorithms for optimizing label-wise metrics budgeted at k that are defined on marginal conditional probabilities of labels.**

The following points describe our main contributions.

1.5.1 Bayes (optimal) classifiers

We derive the form of the Bayes (optimal) classifier for the family of weighted instance-wise metrics that include precision@k or Hamming score@k, and their weighted variants, such as their propensity-scored versions. Following the literature, we include the results for recall@k [Menon et al., 2019] and (n)DCG [Jasinska and Dembczyński, 2018]. We do the same for label-wise metrics, including macro-averaged metrics under expected test utility (ETU) and population utility (PU) frameworks. Our findings demonstrate that across all considered metrics, the Bayes classifier can be defined based on marginal conditional probabilities of labels, allowing for the optimization of all these metrics using plug-in inference algorithms on top of existing label probability estimators. Considering that XMLC requires efficient algorithms, if necessary, we construct computationally efficient approximations that work in linear time with respect to the number of labels. For instance-wise metrics, an optimal decision rule boils down to simple reweighting of labels' probabilities and selection of top-k labels. However, for label-wise metrics, the inference procedure becomes more complex. Because of that, we propose an inference procedure based on the block coordinate ascend (BCA) algorithm for the ETU setting, and the algorithm based on the Frank Wolfe (FW) [Frank and Wolfe, 1956] method for finding the optimal classifier under the PU setting.

1.5.2 Regret bounds

We formally quantify the influence of the estimation error of the labels' marginal conditional probabilities on the suboptimality of the resulting classifier for all proposed inference algorithms. These results are expressed in the form of regret bounds [Bartlett et al., 2006, Narasimhan et al., 2015, Kotłowski and Dembczyński, 2017, Dembczyński et al., 2017]. The notion of consistency varies slightly across different statistical frameworks. A consistent classifier in the standard expected instance-wise utility (EIU) setting optimizes the expected utility on a single sample as the size of the training set increases. Under the ETU framework, the consistent classifier is the one that optimizes the expected prediction utility over a given test set. The PU framework focuses on estimation, in the sense that a consistent PU classifier is one that converges to the population optimal utility as the size of the training set increases. We show that for most metrics considered in this thesis, small estimation errors in label probabilities result in correspondingly small performance degradation, confirming the practical viability of our methods. Furthermore, under the assumption that the underlying method of estimating labels' marginal conditional probabilities is consistent (i.e., the estimation error decreases with increasing training data), our inference methods are also consistent, ensuring theoretical guarantees.

1.5.3 Efficient inference methods

The introduced inference methods have linear complexity in the number of labels, which is considered to be costly in the XMLC setting. Because of that, we demonstrate that by leveraging the sparsity of labels, we can reduce this complexity by orders of magnitude at a small cost of regret. We then examine probabilistic label trees (PLTs) [Jasinska et al., 2016], a popular family of XMLC classifiers that organize labels into a tree structure and factorize label probabilities over tree nodes. Although we also contribute consistency proofs for PLTs and propose their usage as the output layer of a neural network, our primary focus is to introduce PLT-based inference methods for finding exactly the top-k labels with reweighted probabilities characterized by sublinear complexity. We achieve this by applying the BF*(best-first-star) [Pearl, 1984] search algorithm as the PLT tree search procedure during inference.

1.5.4 Empirical evaluation

Finally, we confirm our theoretical findings with extensive empirical experiments on several popular XMLC benchmark datasets for the XMLC repository [Bhatia et al., 2016]. We first simulate the setting of perfect knowledge of the conditional marginal probabilities of labels, eliminating the main source of regret of all the methods and confirming their optimality We then conduct experiments using original labels to reflect the realistic scenario of imperfect probability estimates and the high data sparsity. Additionally, we demonstrate that the introduced methods can be used to optimize label-wise utilities that combine an instance-wise metric, favoring head labels, and a macro-averaged metric. This leads to a classifier that significantly improves tail-label performance with only a tiny sacrifice of head-label performance. Furthermore, this trade-off can be precisely controlled. Finally, we investigate the computational efficiency of a PLT inference method based on BF*-search, showing that it can be computationally less expensive compared to the simpler two-step strategy in which the prediction of marginal probabilities is followed by reweighting.

1.6 Related work

1.6.1 Optimization complex performance metrics

The problem of optimizing complex performance metrics is well-known, with many articles published for a variety of metrics and different classification problems. It has been considered for binary [Ye et al., 2012, Koyejo et al., 2014, Busa-Fekete et al., 2015, Dembczyński et al., 2017], multi-class [Narasimhan et al., 2015, 2022], multi-label [Waegeman et al., 2014, Koyejo et al., 2015, Kotłowski and Dembczyński, 2017], and multi-output [Wang et al., 2019b] classification.

Initially, the main focus was on designing algorithms, without a conscious emphasis on the statistical consequences of choosing models and their asymptotic behavior. Notable examples of such contributions are the SVMperf algorithm [Joachims, 2005], approaches suited to different types of the F-measure [Dembczyński et al., 2011, Natarajan et al., 2016b, Jasinska et al., 2016], or precision at the top [Kar et al., 2015]. The wide use of such complex metrics has caused an increasing interest in investigating their theoretical properties, which can then serve as a guide to design practical algorithms.

The consistency of learning algorithms is a well-established problem. The seminal work of Bartlett et al. [2006] studied this problem for binary classification under the misclassification error. Since then a wide spectrum of learning problems and performance metrics has been analyzed in terms of consistency. These results concern ranking [Duchi et al., 2010, Ravikumar et al., 2011, Calauzenes et al., 2012, Yang and Koyejo, 2020], multi-class classification [Zhang, 2004, Tewari and Bartlett, 2007], multi-label classification [Koyejo et al., 2015, Kotłowski and Dembczyński, 2017] classification with abstention [Yuan and Wegkamp, 2010, Ramaswamy et al., 2018], or constrained classification problems [Agarwal et al., 2018, Kearns et al., 2018].

Our contribution is close to [Koyejo et al., 2015] and [Kotłowski and Dembczyński, 2017]. These articles analyze theoretically complex performance measures known from binary classification in the context of macro- and micro-averages in multi-label classification. However, they do not consider the predictions "at k".

Narasimhan et al. [2015] consider optimization of complex performance measures in multi-class setting and PU framework. In the follow-up article [Narasimhan et al., 2022], they extend the analysis to constraints defined by arbitrary functions of the confusion matrix. These constraints are, however, again different than the budgeted predictions in our case. We suspect that their approach can be generalized to our setting. And our Frank Wolfe algorithm was inspired by theirs.

On the other end of the spectrum are the budgeted at k instance-wise and micro-averaged metrics, such as precision@k or recall@k. The optimal classifiers for these metrics boil down to solving independent binary or multi-class problems via reduction like one-vs-all, pick-all-labels, pick-one-labels [Menon et al., 2019].

1.6.2 Efficient XMLC methods

In this thesis, we address not only the problem of making optimal predictions at k but also ensuring computational efficiency in the XMLC setting. Specifically, the inference methods we discuss can integrate efficiently with any label probability estimator capable of rapidly identifying a subset of top k', where $k' > k$, labels based on predicted marginal conditional probabilities of labels or scores that can be calibrated to probabilities. To provide context for our work, we present a brief overview of relevant methodologies in this domain. Please note that we provide here merely one simplified categorization among many possible perspectives on XMLC methods.

The past decade has witnessed the emergence of many diverse methods, which

aim to reduce computational costs and enhance scalability in XMLC problems. These include 1-vs-all-based approaches that leverage label sparsity [Yen et al., 2017, Babbar and Schölkopf, 2017], sophisticated label filtering methods [Vijayanarasimhan et al., 2014, Shrivastava and Li, 2015, Niculescu-Mizil and Abbasnejad, 2017], decision tree-based algorithms [Prabhu and Varma, 2014, Choromanska and Langford, 2015], and coding-based reduction strategies [Jasinska and Karampatziakis, 2016, Evron et al., 2018, Medini et al., 2019]. Recently, two families of methods have come to dominate the XMLC landscape: label tree methods and label embedding-based methods.

The label tree methods organize labels into a hierarchical tree structure, with individual labels placed at the leaves of this tree. Modern variants typically construct this tree via hierarchical clustering of labels, continuing until each cluster contains fewer than a few hundred labels. The tree structure over final label clusters is often called a router as additional classifiers in the inner nodes of the tree are tasked with guiding the inference into relevant clusters of labels. Classically, label tree approaches relied on sparse TF-IDF features [Leskovec et al., 2014] and trained separate linear classifiers at each tree node [Jasinska et al., 2016, Prabhu et al., 2018b, Khandagale et al., 2020, Jasinska-Kobus et al., 2020, Yu et al., 2022]. Over time, these models have evolved to incorporate more expressive architectures: starting with word and feature embeddings combined with a tree with linear models serving as both output and loss layer [Wydmuch et al., 2018], followed by LSTM-based models [Hochreiter and Schmidhuber, 1997] combined with an attention mechanism [Lin et al., 2017] in each tree node [You et al., 2019b]. Most recently, encoder-only transformer architectures [Vaswani et al., 2017, Devlin et al., 2019] have been used to generate contextual representations of input instances [Chang et al., 2020, Ye et al., 2020, Zhang et al., 2021, Kharbanda et al., 2022a]. Notably, label tree methods operate under a standard classification framework and do not require any additional metadata or features for the labels beyond the training set of instance-labels samples. In this work, we discuss and extend probabilistic label trees [Jasinska et al., 2016], which are a special kind of label trees that estimate marginal conditional probabilities of labels by decomposing them on the paths from root to leaf in the tree.

A parallel line of research to label trees is focusing on label embedding-based methods. These approaches assume that the dimensionality of the label space has an implicit low-rank structureand seek to embed the feature space and label space into a joint low-dimensional space. Initial embedding-based approaches utilized simple linear projections [Mineiro and Karampatziakis, 2015, Yu et al., 2014, Bhatia et al., 2015]. Recognizing their limited expressiveness, subsequent approaches incorporated nonlinear transformations [Tagami, 2017b], and naturally started leveraging deep learning architectures [Yeh et al., 2017, Zhang et al., 2018, Wang et al., 2019a, Guo et al., 2019] utilizing autoencoders [Vincent et al., 2010] and dual encoders [Gillick et al., 2018]. For the inference, these deep learning-based methods rely on approximate nearest neighbor search algorithms [Malkov and Yashunin, 2018, Jayaram Subramanya et al., 2019]. The recent methods started taking advantage of the fact that labels often come with additional features like natural

language name or description, incorporating them through architectures like graph neural networks [Kipf and Welling, 2016] and encoder-only transformers [Zong and Sun, 2020, Saini et al., 2021]. To further enhance the predictive performance of embedding-based methods, they are often combined with additional per-label classifiers [Dahiya et al., 2021, Mittal et al., 2021, Saini et al., 2021, Dahiya et al., 2023a,b, Gupta et al., 2023].

The inference algorithms we introduce in this work are directly applicable to most of these XMLC methods. They work as plug-in approaches, relying on the estimates of the marginal conditional probabilities of labels. Most of the methods listed above either directly estimate these probabilities or predict a per-label score that can be calibrated. Furthermore, our approach can be combined with algorithmic approaches for dealing with missing labels [Saito et al., 2020, Qaraei et al., 2021] as well as with recently proposed methods that try to improve predictive performance on the tail labels by, e.g., leveraging label features to estimate label correlations [Mittal et al., 2021, Saini et al., 2021, Dahiya et al., 2021, Zhang et al., 2022] or to do data augmentation and generate new training points for tail labels [Wei et al., 2021, Kharbanda et al., 2022b, Chien et al., 2023].

1.7 Outline

This dissertation is organized into the following chapters:

- In Chapter 2, we formally introduce the setting of multi-label classification from the point of view of statistical decision theory. We then introduce the concept of performance metrics based on confusion matrices and introduce two frameworks under which they can be optimized: expected test utility (ETU) and Population Utility (PU). Then, we discuss the specific characteristics of XMLC, including long-tailed label distributions, prediction with the "at k" budget constraint, and the problem of missing labels.
- In Chapter 3, we analyze popular instance-wise metrics used in XMLC, including precision@k, recall@k, and DCG@k, as well as their weighted variants. For each metric, we derive the form of the optimal classifier and provide regret bounds in terms of the probability estimation errors.
- In Chapter 4, we extend the analysis of instance-wise metrics to the setting with missing (noisy) labels. We then critically examine the popular propensity model of Jain et al. [2016] (a model that estimates the probabilities that labels are missing). We also discuss the relationship between metrics using propensity models (so-called propensity-scored metrics) to estimate true performance on noisy data and tail label performance, motivating the need for developing and popularizing new metrics focusing on the performance of rare labels.
- In Chapter 5, we introduce label-wise utilities, and especially macro-averaged metrics, that belong to a family of performance metrics based on confusion

matrices and can serve as a good alternative for evaluating tail-label performance. We demonstrate that optimization of label-wise utilities under prediction constrained at k is a non-trivial optimization problem.

- In Chapter 6, we analyze the problem of optimization of label-wise metrics with budget $@k$ under the ETU framework and show the form of the optimal classifier. We then propose an efficient approximate inference procedure based on the block coordinate ascent (BCA) algorithm and provide regret bounds.
- In Chapter 7, analogously to the previous chapter, we analyze the problem of optimization of label-wise metrics under the PU framework. We show the form of the optimal classifier, propose an algorithm for finding the optimal classifier using the Frank-Wolfe (FW) method, and provide regret bounds.
- In Chapter 8, we discuss efficient implementation strategies. First, we present a general method that leverages the sparsity of the label space to reduce the computational complexity of the introduced methods. Then, we introduce the probabilistic label tree (PLT), a popular classifier in XMLC, and show how to use its hierarchical structure for efficient inference for the discussed methods.
- In Chapter 9, we present empirical experiments that validate our theoretical findings and compare all algorithms introduced under the unified protocol.
- In Chapter 10, we conclude this dissertation with a brief summary.
- In Appendix A, we present the full derivation of the proofs that we found to be too long to present in the main body of this thesis.
- In Appendix B, we discuss additional technical details regarding empirical experiments and additional results.

2
Problem setting

In this chapter, we formally define the setting of this dissertation, starting from the notation of multi-label classification problems. We focus on the statistical decision theory point of view and define the risk of a classifier, the Bayes optimal classifier, and the regret of a classifier for instance-wise performance metrics and the general family of performance metrics defined on the confusion matrix. For the latter, we introduce two frameworks: population utility (PU) and expected test utility (ETU), under which we will analyze these metrics in the following chapters. Finally, we introduce the setting of extreme multi-label classification, with its specific properties, which include: long-tailed label distribution, classifiers budgeted at k position, and missing labels that we characterize using the propensity model.

2.1 Multi-label classification

The goal of standard multi-label classification is to find a function mapping between instances $\boldsymbol{x} \in \mathcal{X}$ (usually represented as a real numbers vector of size d, then $\mathcal{X} := \mathbb{R}^d$) and a finite set of m non-mutually-exclusive labels. This means that any $\boldsymbol{x}$ is associated with a subset $\mathcal{L}(\boldsymbol{x}) \subseteq [m]$ of the labels called the relevant or positive labels, with the complement, $[m] \backslash \mathcal{L}(\boldsymbol{x})$, of the irrelevant or negative labels. The set of positive labels might be empty. We identify the relevant labels with a binary vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_m] \in \mathcal{Y}$ in which $y_j = \mathbb{1}[j \in \mathcal{L}(\boldsymbol{x})]$,where $\mathcal{Y} := \{0, 1\}^m$ is called the label vector space. We assume that observations $(\boldsymbol{x}, \boldsymbol{y})$ are generated independently and identically (i.i.d.) according to a (unknown) joint probability distribution of samples $\mathbb{P}[\boldsymbol{x}, \boldsymbol{y}]$ on $\mathcal{X} \times \mathcal{Y}$. We also sometimes refer to that distribution as population distribution. Notice that the above definition concerns not only multi-label classification but also multi-class (when $\|\boldsymbol{y}\|_1 = 1$ for all $\boldsymbol{x}$) and binary classification (when $m = 1$) as special cases.

Generally, we want to find a classifier $\boldsymbol{h} : \mathcal{X} \longrightarrow \mathcal{Y}$, which is a function that

returns prediction vector $\widehat{\boldsymbol{y}} := \boldsymbol{h}(\boldsymbol{x})$ for each instance $\boldsymbol{x}$, that achieves the best possible expected value of a given metric of performance. To construct such a classifier, one is usually given a collection of i.i.d. samples (tuples of corresponding $\boldsymbol{x}$ and $\boldsymbol{y}$) $\mathcal{D}_{\text{train}} := [(\boldsymbol{x}_i, \boldsymbol{y}_i)]_{i=1}^{n_{\text{train}}}$ of size n_{train}, we refer to this collection as a training set[1].

In this work, we consider many different performance metrics. We use the term task utility to refer to a metric that we want to maximize and the term task loss to refer to a metric that we want to minimize. Any loss function can be converted to a utility function by taking its negative or subtracting it from its max/min value, and vice versa. For the consistency of the analysis, throughout this work, we mostly talk about maximizing the task utilities, even if the given performance metric is more popular in its loss variant.

Depending on the form of the task utility, we consider different frameworks to formally characterize the problem. Let us start with the most common one, that is used for popular family of instance-wise utilities (e.g., precision@k, and recall@k). Utilities of this type can be decomposed into the average of individual and independent evaluations for each instance. For these utilities we define the expected instance-wise utility (EIU) of a classifier as:

$$\Phi_{\text{EIU}}(\boldsymbol{h}) := \mathbb{E}_{\boldsymbol{x},\boldsymbol{y} \sim \mathbb{P}[\boldsymbol{x},\boldsymbol{y}]} \left[\Psi(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x}))\right] , \tag{2.1}$$

where $\Psi(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x}))$ is an instance-wise task utility function of a true labels vector $\boldsymbol{y}$ and classifier's prediction $\boldsymbol{h}(\boldsymbol{x})$[2]. Notice that the expected utility corresponds to the popular concepts of risk or generalization error, which are defined using loss functions instead of utilities. The optimal (Bayes) classifier $\boldsymbol{h}^\star_{\text{EIU}}$ is defined as:

$$\boldsymbol{h}^\star_{\text{EIU}} \in \operatorname*{arg\,max}_{\boldsymbol{h} \in \mathcal{H}} \Phi_{\text{EIU}}(\boldsymbol{h}) . \tag{2.2}$$

Please note that we use here the "$\in$" sign instead of ":=" with $\arg\max$ as it is possible that more than one optimal classifier exists. For example, if we consider some budget-constrained metrics like precision@k and recall@k, which are the main focus of this work, then different predictions $\widehat{\boldsymbol{y}}$ can be optimal. We denote the prediction of such optimal classifier as $\boldsymbol{y}^\star := \boldsymbol{h}^\star(\boldsymbol{x})$.

Using the expected utility of the classifier Φ_{EIU} and the corresponding optimal classifier $\boldsymbol{h}^\star_{\text{EIU}}$, we define the regret of a classifier $\boldsymbol{h}$ as:

$$\text{Reg}_{\text{EIU}}(\boldsymbol{h}) := \Phi_{\text{EIU}}(\boldsymbol{h}^\star_{\text{EIU}}) - \Phi_{\text{EIU}}(\boldsymbol{h}) . \tag{2.3}$$

It measures the susceptibility of $\boldsymbol{h}$ with respect to Ψ. From the definition, we have that $\text{Reg}(\boldsymbol{h}) \geqslant 0$ for every classifier $\boldsymbol{h}$, and $\text{Reg}(\boldsymbol{h}) = 0$ if and only if $\boldsymbol{h}$ is optimal. If the regret of a classifier $\boldsymbol{h}$ converges to zero with the training sample size tending to infinity $n_{\text{train}} \to \infty$, we call such an algorithm (statistically) consistent. We can

[1] Despite the "set" naming convention, $\mathcal{D}$ is formally defined as vector, as it may contain the same tuple of $(\boldsymbol{x}, y)$ more than once and some algorithms may take into account the order of samples in $\mathcal{D}$.

[2] In this work, we generally use Ψ to denote utility as a function of true labels and predictions of a classifier, and Φ to denote different expected values of utility functions.

now formally define the goal of multi-label classification. It is to find a classifier $\boldsymbol{h}$ with the smallest possible regret $\mathrm{Reg}(\boldsymbol{h})$ under a given framework and task utility.

Empirically, the utility of the classifier $\boldsymbol{h}$ is estimated using a given collection of instances $\mathcal{D} := [(\boldsymbol{x}_i, \boldsymbol{y}_i)]_{i=1}^n$ of a given size n, that is assumed to be sampled i.i.d. from $\mathbb{P}[\boldsymbol{x}, \boldsymbol{y}]$, we refer to it as a test set. A test set can also be seen as a tuple $(\mathbf{X}, \mathbf{Y})$, where $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$ is an $n \times d$ matrix of instances, and $\mathbf{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n]$ is an $n \times m$ matrix of labels. We will use this matrix notation more often due to its simplicity. Following this convention, we use $\boldsymbol{x}_i$ to denote i-th row of $\mathbf{X}$, $\boldsymbol{y}_i$ to denote i-th row of $\mathbf{Y}$ (a vector of all labels for a single instance i), $\boldsymbol{y}_{:,j}$ to denote j-th column of $\mathbf{Y}$ (a vector single label j for all instance), and $y_{i,j}$ to denote a single label j for a single instance i. To complement this notation, we will use a classifier on $\mathbf{X}$ to denote its prediction for all instances $\boldsymbol{x}_i$ in $\mathbf{X}$: $\widehat{\mathbf{Y}} := \boldsymbol{h}(\mathbf{X}^{n\times d}) = [\boldsymbol{h}(\boldsymbol{x}_i)]_{i=1}^n$, and the prediction of optimal classifier $\mathbf{Y}^\star := \boldsymbol{h}^\star(\mathbf{X})$.

2.2 Generalized performance metrics

In this work, we analyze not only the instance-wise performance metrics but also generalized performance metric for multi-label classification that can be non-decomposable over instances, meaning they cannot be evaluated for each instance independently and then averaged. This property has led to two distinct frameworks for characterizing and analyzing the problem of optimizing such metrics. In this section, we introduce the notation of the confusion matrix, which is used to define a wide family of generalized performance metrics. Then we define these two frameworks, which are commonly known as population utility (PU) [Koyejo et al., 2014, Narasimhan et al., 2014] and expected test utility (ETU) [Dembczyński et al., 2012, Waegeman et al., 2014, Natarajan et al., 2016a].

2.2.1 Multi-label confusion matrix

We start with the definition of a confusion matrix for multi-label classification. Given vectors of true labels $\boldsymbol{y} \in \{0,1\}^n$ and predicted labels $\widehat{\boldsymbol{y}} \in \{0,1\}^n$, let us define the entries of binary confusion matrix, named as true positives (tp), true negatives (tn), false positives (fp), false negatives (fn) ratios:

$$\begin{aligned} \mathrm{tn}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) &:= \frac{1}{n}\sum_{i=1}^n (1-y_i)(1-\widehat{y}_i)\,, & \mathrm{fp}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) &:= \frac{1}{n}\sum_{i=1}^n (1-y_i)\widehat{y}_i\,, \\ \mathrm{fn}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) &:= \frac{1}{n}\sum_{i=1}^n y_i(1-\widehat{y}_i)\,, & \mathrm{tp}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) &:= \frac{1}{n}\sum_{i=1}^n y_i\widehat{y}_i\,. \end{aligned} \tag{2.4}$$

The binary confusion matrix is then a vector composed from these entries $\widehat{\boldsymbol{c}}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = [\mathrm{tn}(\boldsymbol{y}, \widehat{\boldsymbol{y}}), \mathrm{fp}(\boldsymbol{y}, \widehat{\boldsymbol{y}}), \mathrm{fn}(\boldsymbol{y}, \widehat{\boldsymbol{y}}), \mathrm{tp}(\boldsymbol{y}, \widehat{\boldsymbol{y}})]$[3]. Notice that in the presented definition, all

[3]For confusion matrices, instead of addressing specific elements with the index, we use instead $\mathrm{tn}, \mathrm{fp}, \mathrm{fn}$, and tp for clarity to which element we refer. We use $\widehat{\boldsymbol{c}}$ as we reserve $\boldsymbol{c}$ for expected value of confusion matrix, which we will use in the later chapters

entries of the binary confusion matrix are normalized to the range $[0, 1]$ and sum up to 1. We define a set of binary confusion matrices $\mathcal{C} = \left\{\widehat{\boldsymbol{c}} \in [0,1]^4 : \|\widehat{\boldsymbol{c}}\|_1 = 1\right\}$. The binary confusion matrix can be computed either for all multi-label predictions for a given single instance $(\boldsymbol{y}_i, \widehat{\boldsymbol{y}}_i)$ or binary predictions for label j obtained on a set of different instances $(\boldsymbol{y}_{:,j}, \widehat{\boldsymbol{y}}_{:,j})$. In the following, we focus on the latter, and we define multi-label confusion matrix as $\widehat{\mathbf{C}}\left(\mathbf{Y}, \widehat{\mathbf{Y}}\right) \in \mathcal{C}^m$:

$$\begin{aligned}\widehat{\mathbf{C}}\left(\mathbf{Y}, \widehat{\mathbf{Y}}\right) &\coloneqq \left[\begin{array}{cccc} \widehat{\boldsymbol{c}}(\boldsymbol{y}_{:,1}, \widehat{\boldsymbol{y}}_{:,1}), & \widehat{\boldsymbol{c}}(\boldsymbol{y}_{:,2}, \widehat{\boldsymbol{y}}_{:,2}), & \dots, & \widehat{\boldsymbol{c}}(\boldsymbol{y}_{:,m}, \widehat{\boldsymbol{y}}_{:,m}) \end{array}\right]^\intercal \\ &= \begin{bmatrix} \mathrm{tn}(\boldsymbol{y}_{:,1}, \widehat{\boldsymbol{y}}_{:,1}) & \mathrm{fp}(\boldsymbol{y}_{:,1}, \widehat{\boldsymbol{y}}_{:,1}) & \mathrm{fn}(\boldsymbol{y}_{:,1}, \widehat{\boldsymbol{y}}_{:,1}) & \mathrm{tp}(\boldsymbol{y}_{:,1}, \widehat{\boldsymbol{y}}_{:,1}) \\ \mathrm{tn}(\boldsymbol{y}_{:,2}, \widehat{\boldsymbol{y}}_{:,2}) & \mathrm{fp}(\boldsymbol{y}_{:,2}, \widehat{\boldsymbol{y}}_{:,2}) & \mathrm{fn}(\boldsymbol{y}_{:,2}, \widehat{\boldsymbol{y}}_{:,2}) & \mathrm{tp}(\boldsymbol{y}_{:,2}, \widehat{\boldsymbol{y}}_{:,2}) \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{tn}(\boldsymbol{y}_{:,m}, \widehat{\boldsymbol{y}}_{:,m}) & \mathrm{fp}(\boldsymbol{y}_{:,m}, \widehat{\boldsymbol{y}}_{:,m}) & \mathrm{fn}(\boldsymbol{y}_{:,m}, \widehat{\boldsymbol{y}}_{:,m}) & \mathrm{tp}(\boldsymbol{y}_{:,m}, \widehat{\boldsymbol{y}}_{:,m}) \end{bmatrix}. \end{aligned} \tag{2.5}$$

The confusion matrix can be computed for a single sample. In such situation, we will use notation $\widehat{\mathbf{C}}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) \coloneqq \widehat{\mathbf{C}}([\boldsymbol{y}], [\widehat{\boldsymbol{y}}])$. Later in the thesis, we often omit the arguments of the confusion matrix or its entities for readability reasons. We say that the performance metric is defined on a confusion matrix if it can be expressed as a function of a confusion matrix:

$$\Psi\left(\mathbf{Y}, \widehat{\mathbf{Y}}\right) = \Psi\left(\widehat{\mathbf{C}}\left(\mathbf{Y}, \widehat{\mathbf{Y}}\right)\right) = \Psi\left(\widehat{\boldsymbol{c}}_{:,\mathrm{tn}}, \widehat{\boldsymbol{c}}_{:,\mathrm{fp}}, \widehat{\boldsymbol{c}}_{:,\mathrm{fn}}, \widehat{\boldsymbol{c}}_{:,\mathrm{tp}}\right). \tag{2.6}$$

Since the entries of the confusion matrix are linearly dependent, it suffices to use three independent combinations of its entries, such as:

$$\begin{aligned} \mathrm{tp}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) &\coloneqq \frac{1}{n}\sum_{i=1}^{n} y_i \widehat{y}_i\,, \\ \mathrm{pp}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) &\coloneqq \mathrm{fp}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) + \mathrm{tp}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = \frac{1}{n}\sum_{i=1}^{n} \widehat{y}_i\,, \\ \mathrm{cp}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) &\coloneqq \mathrm{fn}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) + \mathrm{tp}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = \frac{1}{n}\sum_{i=1}^{n} y_i\,, \end{aligned} \tag{2.7}$$

where tp is the same ratio of true positives, pp is the ratio of predicted positives, and cp is the ratio of (conditionally) positive labels. Notice that pp if fact depends only on $\widehat{\boldsymbol{y}}$, while the cp only on $\boldsymbol{y}$. We will use this alternative parameterization of the confusion matrix when it is more convenient for the analysis:

$$\Psi\left(\widehat{\mathbf{C}}\left(\mathbf{Y}, \widehat{\mathbf{Y}}\right)\right) = \Psi\left(\widehat{\boldsymbol{c}}_{:,\mathrm{tp}}, \widehat{\boldsymbol{c}}_{:,\mathrm{pp}}, \widehat{\boldsymbol{c}}_{:,\mathrm{cp}}\right). \tag{2.8}$$

The above form of confusion matrix is not the only one possible. Another popular form for multi-label classification is defined over all possible vectors in $\mathcal{Y}$ with entries $\widehat{\mathbf{C}}_{\boldsymbol{y},\widehat{\boldsymbol{y}}}$ being equal to the ratio of samples with label vector $\boldsymbol{y}$ and prediction $\widehat{\boldsymbol{y}}$. This is however not practical in the XMLC setting, as the number of entries in such matrix is 2^{2m}.

On the other hand, the introduced confusion matrix is more compact and the binary confusion matrix available for each label naturally allows to define utility functions that balance the contributions of different labels to the final metric.

2.2.2 Two frameworks for generalised performance metrics

In this work, we will consider two frameworks for characterizing the performance metrics, commonly known as population utility (PU) [Koyejo et al., 2014, Narasimhan et al., 2014] and expected test utility (ETU) [Dembczyński et al., 2012, Waegeman et al., 2014, Natarajan et al., 2016a]. Here we introduce both of them before getting into the further discussion on the performance metrics and optimal classifiers in Chapters 3 and 5.

2.2.3 Expected test utility (ETU)

In the expected test utility (ETU) framework (also known in the literature as decision theoretic analysis (DTA) [Ye et al., 2012]) the goal is to find a classifier with the best expected utility on a given test set of instances $\mathbf{X}$. In the ETU framework, the expected utility of classifier $\Phi_{\mathrm{ETU}}(\boldsymbol{h})$ is defined as:

$$\Phi_{\mathrm{ETU}}(\boldsymbol{h}) \coloneqq \mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \,|\, \mathbf{X}]} \left[\Psi \left(\widehat{\mathbf{C}}(\mathbf{Y}, \boldsymbol{h}(\mathbf{X}) \right) \right] , \tag{2.9}$$

and optimal (Bayes) classifier $\boldsymbol{h}^{\star}_{\mathrm{ETU}}$ is defined as:

$$\boldsymbol{h}^{\star}_{\mathrm{ETU}} \in \operatorname*{arg\,max}_{\boldsymbol{h} \in \mathcal{H}} \Phi_{\mathrm{ETU}}(\boldsymbol{h}) \,, \tag{2.10}$$

and regret of a classifier $\boldsymbol{h}$ as:

$$\mathrm{Reg}_{\mathrm{ETU}}(\boldsymbol{h}) \coloneqq \Phi_{\mathrm{ETU}}(\boldsymbol{h}^{\star}_{\mathrm{ETU}}) - \Phi_{\mathrm{ETU}}(\boldsymbol{h}) \,. \tag{2.11}$$

We can interpret this setting as optimizing the expected metric over an infinite set of realizations of the test set, with the same instances $\mathbf{X}$ but different labels $\mathbf{Y}$ drawn from distribution $\mathbb{P}[\mathbf{Y} \,|\, \mathbf{X}]$. Here the classifier $\boldsymbol{h}$ predicts for the whole set of instances $\mathbf{X}$ (set-wise). Notice that the i.i.d. assumption implies that the labels $\boldsymbol{y}_i$ corresponding to $\boldsymbol{x}_i$ do not depend on any other instance: $\mathbb{P}[\mathbf{Y} \,|\, \mathbf{X}] = \prod_{i=1}^{n} \mathbb{P}[\boldsymbol{y}_i \,|\, \boldsymbol{x}_i]$.

Framing a problem like that can be helpful in real life when, ahead of time, the prediction system is given a specific batch of instances for which it has to compute the predictions all at once, e.g., a recommendation system that, during the night, predicts recommendations for all the users that will be displayed for the next day.

Note that the ETU framework does not require a utility to be defined on the confusion matrix but only on the labels and predictions. However, to make the difference with the PU framework clearer, and overall notation more consistent, we define it here using the utilities defined on a confusion matrix.

2.2.4 Population utility (PU)

In the population utility (PU) framework (also known as empirical utility maximization (EUM) [Ye et al., 2012]), the goal is to optimize a performance metric calculated on expected value of a confusion matrix over the population distribution $\mathbb{P}[\boldsymbol{x}, \boldsymbol{y}]$. In this framework, the expected utility of classifier $\Phi_{\mathrm{PU}}(\boldsymbol{h})$ is defined as:

$$\Phi_{\mathrm{PU}}(\boldsymbol{h}) \coloneqq \Psi\left(\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}\left[\widehat{\mathbf{C}}(\boldsymbol{y},\boldsymbol{h}(\boldsymbol{x}))\right]\right), \tag{2.12}$$

with optimal (Bayes) classifier $\boldsymbol{h}^{\star}_{\mathrm{PU}}$ is defined as:

$$\boldsymbol{h}^{\star}_{\mathrm{PU}} \in \operatorname*{arg\,max}_{\boldsymbol{h}\in\mathcal{H}} \Phi_{\mathrm{PU}}(\boldsymbol{h}), \tag{2.13}$$

and regret of a classifier $\boldsymbol{h}$ as:

$$\mathrm{Reg}_{\mathrm{PU}}(\boldsymbol{h}) := \Phi_{\mathrm{PU}}(\boldsymbol{h}^{\star}_{\mathrm{PU}}) - \Phi_{\mathrm{PU}}(\boldsymbol{h}). \tag{2.14}$$

In other words, we can understand this setting as optimizing the utility over a single infinitely large test set, where classifier $\boldsymbol{h}$ predicts for each instance $\boldsymbol{x}$ independently (point-wise).

The PU setting is clearly practical when the prediction system is designed to provide predictions for a continuous stream of instances without being able to change its previous predictions or even have any memory of them.

2.2.5 Relationship between ETU and PU frameworks

The ETU and PU frameworks are equivalent for some performance metrics in the sense that the optimal classifiers for both frameworks produce the same set of optimal predictions. This is true for utilities that are linear w.r.t. the confusion matrix (e.g., instance-wise metrics)[4]:

$$\Psi\left(\mathbb{E}_{\widehat{\mathbf{C}}}\left[\widehat{\mathbf{C}}\right]\right) = \mathbf{G}\cdot\mathbb{E}_{\widehat{\mathbf{C}}}\left[\widehat{\mathbf{C}}\right] = \mathbb{E}_{\widehat{\mathbf{C}}}\left[\mathbf{G}\cdot\widehat{\mathbf{C}}\right] = \mathbb{E}_{\widehat{\mathbf{C}}}\left[\Psi\left(\widehat{\mathbf{C}}\right)\right]. \tag{2.15}$$

Because of the i.i.d. assumption, it is clear that, for such utilities, the optimal classifier has the same form that independently predicts for each instance.

$$\boldsymbol{h}^{\star}_{\mathrm{PU}}(\boldsymbol{x}) = \boldsymbol{h}^{\star}_{\mathrm{ETU}}(\boldsymbol{x}) \tag{2.16}$$

While the class of simple instance-wise performance metrics is definitely the most popular in machine learning, the aim of this work is to take a closer look at macro-averaged performance metrics, which are usually non-decomposable w.r.t. instances, i.e., nonlinear w.r.t. the confusion matrix, and therefore the optimal classifiers for ETU and PU frameworks are different, forcing us to analyze the problem from both perspectives, as they clearly have their advantages and disadvantages in modeling different problems.

[4] We use $\boldsymbol{a}\cdot\boldsymbol{b} = \sum_{i=1}^{n} a_i b_i$ to denote the dot product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ of size n, and $\mathbf{A}\cdot\boldsymbol{B} = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{i,j} b_{i,j}$ to denote the matrix dot product of $\mathbf{A}$ and $\mathbf{B}$ of size $n\times m$.

It is also worth noting that, under the i.i.d. assumption and for some utility functions Ψ, $\Phi_{\mathrm{ETU}}(\boldsymbol{h})$ converges to $\Phi_{\mathrm{PU}}(\boldsymbol{h})$ when the size n of the test set $\mathbf{X}, \mathbf{Y}$ is going to infinity, meaning they are asymptotically equivalent. Under the i.i.d. assumption $\widehat{\mathbf{C}}\left(\mathbf{Y}, \boldsymbol{h}(\mathbf{X})\right)$ converges to $\mathbb{E}\left[\widehat{\mathbf{C}}\left((\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})\right)\right]$, as $n \to \infty$. Let us denote the confusion matrix calculated on $\mathbf{X}, \mathbf{Y}$ of size n as $\widehat{\mathbf{C}}^n$. If Ψ is bounded, as all the utilities we consider in this work, we can exchange the limit with the expectation:

$$\lim_{n\to\infty} \mathbb{E}\left[\Psi\left(\widehat{\mathbf{C}}^n\right)\right] = \mathbb{E}\left[\lim_{n\to\infty} \Psi\left(\widehat{\mathbf{C}}^n\right)\right] = \Psi\left(\mathbb{E}\left[\widehat{\mathbf{C}}\right]\right) \tag{2.17}$$

The more detailed proofs of this asymptotic equivalence that additionally derive the convergence rate can be found in Ye et al. [2012], Dembczyński et al. [2017].

2.3 Specificity of extreme multi-label classification setting

Extreme multi-label classification (XMLC) differs from classic multi-label classification. In this section, we will focus on specific properties of XMLC that change classical multi-label classification settings.

2.3.1 Long-tailed label distribution

The defining characteristic of extreme classification datasets is the very large number of labels m reaching up to hundreds of thousands or even millions (we usually call the problem *extreme* when $m \geqslant 10^4$). The number of relevant labels for the sample, $\|\boldsymbol{y}\|_1$, is usually much smaller than m ($\|\boldsymbol{y}\|_1 \ll m$). This causes both computational and statistical challenges. The obvious computational challenge is that, with such a large number of labels , learning and prediction demand a lot of computational resources. Because of that, XMLC requires specialized algorithms that work in sublinear time w.r.t. the number of labels m. The statistical challenge, that is directly connected to the high dimensionality of the problem and the small number of relevant labels per sample, is the label distribution being usually highly imbalanced.

In the case of binary classification, the amount of imbalance is completely determined by the imbalance ratio $\frac{\mathbb{P}[y=0]}{\mathbb{P}[y=1]}$. In this sense, almost every binary problem corresponding to predicting the relevance of a single label is highly imbalanced in XMLC, i.e., only a small fraction of training instances will be associated with that label.

However, in XMLC, the data are also imbalanced when comparing different labels. And because of that, we can also define an inter-label imbalance ratio (ILIR). Let us denote the prior probability of label j as $\pi_j := \mathbb{P}[y_i = 1]$, then $\mathrm{ILIR} := \frac{\max\{\pi_i : i \in [m]\}}{\min\{\pi_j : j \in [m]\}}$. This factor is also usually very high, as some labels have a large number of positives, while most of the labels have only very few. This described type of label distribution that we encounter in XMLC problems is said to be long-tailed [Bhatia et al., 2015, Babbar and Schölkopf, 2017]. And so, we

refer to these less frequent labels as tail labels and to the more frequent ones as head labels.

Under the long-tailed label distribution, we assume that predicting tail labels is more rewarding than predicting head ones, which is reflected by aiming to optimize the performance metrics that emphasize the correct prediction of tail labels. In Chapters 5 to 7, we will discuss such metrics and ways to optimize them.

2.3.2 Classifier budgeted at k position

Another distinguishing characteristic of XMLC compared to standard multi-label classification (as defined in Section 2.1) is that applications of the former usually limit the number of labels that can be predicted for each instance. For example, in recommendation and information retrieval systems, we are limited to presenting only k results to a user due to user interface design and limitation. In this context, it is preferable in XMLC to evaluate the classifier's ability to retrieve the top k relevant labels, where k is a predefined budget or cutoff point.

Therefore, we consider a subset of label vectors with exactly k ($k \leqslant m$) relevant labels $\mathcal{Y}^{@k} = \{\boldsymbol{y} \in \mathcal{Y} : \|\boldsymbol{y}\|_1 = k\}$, and a class of classifiers budgeted at k position $\mathcal{H}^{@k} = \{\boldsymbol{h} \in \mathcal{H} : h_j(\boldsymbol{x}) \in \mathcal{Y}^{@k}, \forall \boldsymbol{x} \in \mathcal{X}\}$, that is, classifiers that always predict exactly k labels. We denote a single such classifier as $\boldsymbol{h}^{@k}$.

In this work, we will focus on the performance metrics that are defined on predictions of classifiers budgeted at k position ($\widehat{\boldsymbol{y}}^{@k} \in \mathcal{Y}^{@k}$). The most popular metric of this type is (instance-wise) precision@k followed by recall@k. However, we are more interested in variants of macro-averaged measures that are budgeted at k position, such as macro-precision@k, macro-recall@k, macro-F_β-measure@k, or coverage@k.

To be able to precisely predict k labels, we consider classifiers $\boldsymbol{h}^{@k}$ that are constructed in a two-step process. First, we are interested in building a label probability estimator (LPE) that estimates the marginal conditional probabilities of labels $\boldsymbol{\eta}(\boldsymbol{x}) = [\eta_1(\boldsymbol{x}), \eta_2(\boldsymbol{x}), \ldots, \eta_m(\boldsymbol{x})]$ where

$$\eta_j(\boldsymbol{x}) = \mathbb{P}[y_j = 1 \,|\, \boldsymbol{x}] = \sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}] y_j \,. \tag{2.18}$$

We want to estimate them as accurately as possible, that is, with possibly a small L_1 estimation error

$$|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})| \,, \tag{2.19}$$

or alternatively L_2 error

$$(\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x}))^2 \,, \tag{2.20}$$

where $\widehat{\eta}_j(\boldsymbol{x})$ is an estimate of $\eta_j(\boldsymbol{x})$ coming from LPE. The LPE is usually obtained by learning an estimator on a whole or part of a training set by minimizing proper composite losses [Reid and Williamson, 2010, Agarwal, 2014, Kotłowski and Dembczyński, 2017]. We discuss one of such algorithms in more detail in Section 8.2. We sometimes use notation $\boldsymbol{\eta}(\mathbf{X}^{n \times d}) = [\boldsymbol{\eta}(\boldsymbol{x}_i)]_{i=1}^n$ to denote matrix of conditional probabilities for all instance $\boldsymbol{x}_i$ in $\mathbf{X}$.

In the second step, we use estimates $\widehat{\eta}(\boldsymbol{x})$ to construct a vector of predicted labels $\widehat{\boldsymbol{y}}^{@k}$. It can be a simple rule, like predicting top-k labels with the highest estimated probabilities $\boldsymbol{\eta}_j(\boldsymbol{x})$ as relevant, or a much more complex algorithm that we will discuss later in this work. This statement of the problem is justified by the fact that the optimal classifier for most of the discussed metrics in this thesis can be written as a function of the conditional probabilities of labels $\eta(\boldsymbol{x})$, as we prove later in Chapters 3, 6 and 7.

Usually, classifier $\boldsymbol{h}^{@k}$ is restricted to $\mathcal{Y}^{@k}$. However, some task losses (DCG@k and nDCG@k) discussed in this work correspond to a slightly different task of ranking, and require the classifier to explicitly order the predicted k labels in terms of relevance. From now on, we will often skip the "budgeted at k" part when referring to $\boldsymbol{h}^{@k} \in \mathcal{H}^{@k}$ and $\widehat{\boldsymbol{y}}^{@k} \in \mathcal{Y}^{@k}$, since we mostly focus on this type of classifiers and related metrics. We will distinguish between constrained and unconstrained classifiers if needed or if it will benefit clarity.

2.3.3 Missing labels

In real-world applications, the observed data might not follow the distribution we want to learn about. In the extreme classification literature, it is a popular assumption that the observed labels are only a subset of the truly relevant labels. This is a reasonable assumption since the number of labels is large; it is often infeasible for a human to annotate all the relevant labels for a given instance.

An illustrative example might be a Wikipedia dataset. The content of a Wikipedia article should be matched with a set of categories the article belongs to. There are currently over 2 million categories on Wikipedia, and so it is clear that the original authors and moderators have never checked every single category for each article.[5] On the other hand, each category that has been assigned to an article has been verified by a human to be relevant.

As another example, let us consider the case of recommendation systems, where a human annotator (usually a user of the system) is most likely to select a few relevant labels from the ones proposed by the system or the one that the annotator can think of. And again, the annotator is unlikely to mark a label as relevant for them when it is, in reality, not. Therefore, the labeling error can be assumed to be strongly one-sided: There may be many missing labels, but spurious labels should be uncommon.

To introduce this modification of the problem setting, we contrast ground-truth labels $\boldsymbol{y}$ with those that are actually observed. We use a check symbol to denote the noisy observed labels $\breve{\boldsymbol{y}}$. Then, generally, this setting can be defined as:

$$\mathbb{P}\left[\breve{\boldsymbol{y}} \preceq \boldsymbol{y} \,|\, \boldsymbol{x}\right] = 1\,, \qquad \mathbb{P}\left[\breve{\boldsymbol{y}} \npreceq \boldsymbol{y} \,|\, \boldsymbol{x}\right] = 0\,, \tag{2.21}$$

where $\breve{\boldsymbol{y}} \preceq \boldsymbol{y}$ means that $\breve{y}_j \leqslant y_j$ for all $j \in [m]$, and $\breve{\boldsymbol{y}} \npreceq \boldsymbol{y}$ means that there is at least one label for which $\breve{y}_j > y_j$. Notice that the above equations also cover the noise-free case, as we may have $\mathbb{P}\left[\breve{\boldsymbol{y}} = \boldsymbol{y} \,|\, \boldsymbol{x}\right] = 1$.

[5] If it took a human one second to check a category for an article, then annotating a single article fully would take almost 6 days.

Let $\eta_{\boldsymbol{y}}(\boldsymbol{x}) \coloneqq \mathbb{P}\left[\boldsymbol{y} = \boldsymbol{y} \,|\, \boldsymbol{x}\right]$ and $\breve{\eta}_{\boldsymbol{y}}(\boldsymbol{x}) \coloneqq \mathbb{P}\left[\breve{\boldsymbol{y}} = \boldsymbol{y} \,|\, \boldsymbol{x}\right]$. The relationship between them is:

$$\breve{\eta}_{\breve{\boldsymbol{y}}}(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \mathcal{Y}} p_{\breve{\boldsymbol{y}}}(\boldsymbol{y}, \boldsymbol{x}) \eta_{\boldsymbol{y}}(\boldsymbol{x}) \,, \tag{2.22}$$

where $p_{\breve{\boldsymbol{y}}}(\boldsymbol{y}, \boldsymbol{x}) \coloneqq \mathbb{P}[\breve{\boldsymbol{y}} = \breve{\boldsymbol{y}} \,|\, \boldsymbol{y} = \boldsymbol{y}, \boldsymbol{x}]$ is a propensity of observing $\breve{\boldsymbol{y}}$ for ground-truth labels $\boldsymbol{y}$ and instance $\boldsymbol{x}$. Notice that from (2.21) we have $p_{\breve{\boldsymbol{y}}}(\boldsymbol{y}, \boldsymbol{x}) = 0$ for $\breve{\boldsymbol{y}} \not\leq \boldsymbol{y}$. Furthermore, let $\boldsymbol{\eta}_{\mathcal{Y}}(\boldsymbol{x})$ and $\breve{\boldsymbol{\eta}}_{\mathcal{Y}}(\boldsymbol{x})$ be vectors of $\eta_{\boldsymbol{y}}(\boldsymbol{x})$ and $\breve{\eta}_{\boldsymbol{y}}(\boldsymbol{x})$, respectively, for all $\boldsymbol{y} \in \mathcal{Y}$ given in some predefined order π. Let $\mathbf{P}$ be a matrix containing all propensities $p_{\breve{\boldsymbol{y}}}(\boldsymbol{y}, \boldsymbol{x})$, with rows and columns corresponding to $\breve{\boldsymbol{y}}$ and $\boldsymbol{y}$, respectively, and organized according to π. Then, we get:

$$\breve{\boldsymbol{\eta}}_{\mathcal{Y}}(\boldsymbol{x}) = \mathbf{P} \boldsymbol{\eta}_{\mathcal{Y}}(\boldsymbol{x}) \,, \tag{2.23}$$

and, finally:

$$\boldsymbol{\eta}_{\mathcal{Y}}(\boldsymbol{x}) = \mathbf{P}^{-1} \breve{\boldsymbol{\eta}}_{\mathcal{Y}}(\boldsymbol{x}) \,, \tag{2.24}$$

where we need to assume that $\mathbf{P}$ is invertible. Therefore, the observed samples can be viewed as generated in a two-step process: first the sample $(\boldsymbol{x}, \boldsymbol{y})$ is generated from the true distribution $\mathbb{P}\left[\boldsymbol{x}, \boldsymbol{y}\right]$ on $\mathcal{X} \times \mathcal{Y}$ and then the observed label vector is drawn according to $\mathbb{P}\left[\breve{\boldsymbol{y}} \,|\, \boldsymbol{y}, \boldsymbol{x}\right]$, resulting in the observed sample $(\boldsymbol{x}, \breve{\boldsymbol{y}})$.

However, reconstruction of the ground-truth distribution from the observed one, in the general case, is not a trivial task from a statistical and computational perspective, as it requires an exponential number of parameters (2^m for a single instance $\boldsymbol{x}$), which is additionally problematic in the extreme classification setting. Because of this practical reason, much simpler, label-wise propensities are commonly used that are defined for each label separately:

$$p_j(\boldsymbol{x}) \coloneqq \mathbb{P}\left[\breve{y}_j = 1 \,|\, y_j = 1, \boldsymbol{x}\right] . \tag{2.25}$$

Let $\breve{\eta}_j(\boldsymbol{x}) \coloneqq \mathbb{P}\left[\breve{y}_j = 1 \,|\, \boldsymbol{x}\right]$ be the marginal conditional probability of the observed label j. Its relation to true $\eta_j(\boldsymbol{x})$ is then:

$$\breve{\eta}_j(\boldsymbol{x}) = p_j(\boldsymbol{x}) \eta_j(\boldsymbol{x}) \,, \qquad \eta_j(\boldsymbol{x}) = \breve{\eta}_j(\boldsymbol{x}) / p_j(\boldsymbol{x}) \,. \tag{2.26}$$

Even simpler variants of this model that are commonly encountered in the literature are label or class-conditional random noise model [Natarajan et al., 2013, Jain et al., 2016, Van Rooyen and Williamson, 2017], where the propensities depends on y but not on the instance $\boldsymbol{x}$:

$$p_j \coloneqq \mathbb{P}\left[\breve{y}_j = 1 \,|\, y_j = 1\right] . \tag{2.27}$$

If propensities are known, then they can be used to construct unbiased utilities or losses $\breve{\Psi}$[Van Rooyen and Williamson, 2017] in the sense that

$$\forall \boldsymbol{h} : \; \Phi_{\mathbb{P}[\boldsymbol{x}, \boldsymbol{y}]}(\boldsymbol{h}) = \breve{\Phi}_{\mathbb{P}[\boldsymbol{x}, \breve{\boldsymbol{y}}]}(\boldsymbol{h}) \,, \tag{2.28}$$

meaning that for every classifier $\boldsymbol{h}$, its expected utility on the true data distribution using the original utility function Ψ is equal to its expected utility on the observed

distribution using the unbiased utility $\breve{\Psi}$. We intentionally present (2.28) in this general form, as the expected utilities of classifier Φ are defined in different ways in the EIU, ETU, and PU frameworks.

The construction of the unbiased counterpart depends on the form of propensities, e.g., the label-wise propensities (2.25) are sufficient for utilities decomposable over labels [Natarajan et al., 2017] (and what we also show in Section 4.1), but might not be for more complex utilities without additional assumptions [Schultheis and Babbar, 2021]. The unbiased utilities can be used in training procedures for the estimator of $\boldsymbol{\eta}$ [Jain et al., 2016, Qaraei et al., 2021] or for estimating the performance of classifiers. We will discuss this topic in more detail in Chapter 4.

Generally, we assume that the task under the missing label assumption is to find the best classifier $\boldsymbol{h}$ for a given utility defined on the true distribution of instances, using the observed (noisy) training set $\breve{\mathcal{D}}_{\text{train}} = [(\boldsymbol{x}_i, \breve{\boldsymbol{y}}_i)]_{i=1}^{n_{\text{train}}}$.

2.4 Summary of the chapter

In this chapter, we have formally defined the problem setting of extreme multi-label classification (XMLC), starting from the fundamental notation of multi-label classification from the perspective of statistical decision theory. We have defined expected utility (corresponding to the classifier's risk), Bayes optimality, and regret, covering both instance-wise performance metrics and generalized metrics defined via confusion matrices. Two primary frameworks for analyzing these generalized metrics, namely Population Utility (PU) and Expected Test Utility (ETU), have been detailed. Further, we have discussed specific characteristics of XMLC problems, including the long-tailed distribution of labels, constraints on classifiers budgeted at position k, and missing labels. Finally, we have formally defined a propensity-based model for addressing label incompleteness.

3 Optimization of instance-wise metrics at k

In this chapter, we focus on the instance-wise metrics budgeted at k, that are commonly used in extreme multi-label classification. The results presented in this chapter include both the contributions of this dissertation and the complementary results from the literature. We start with the most popular measure, namely (weighted) precision@k, and present the form of the corresponding optimal classifier [Wydmuch et al., 2018, 2021]. We then generalize this result to the class of instance-wise weighted utilities that include other popular metrics like (weighted) Hamming loss. We present the same for recall@k [Menon et al., 2019] (whose work was partially inspired by [Wydmuch et al., 2018]), and (n)DCG@k [Jasinska and Dembczyński, 2018], which are other popular instance-wise metrics in XMLC.

3.1 Precision@k

Unquestionably, the most popular metric used in the field of extreme multi-label classification is precision@k, which is defined as a fraction of the number of correct predictions among the k predicted labels:

$$\mathrm{P@}k\left(\boldsymbol{y}, \widehat{\boldsymbol{y}}^{@k}\right) \coloneqq \frac{1}{k}\sum_{j=1}^{m} y_j \widehat{y}_j \,. \tag{3.1}$$

It is also common to use a weighted variant of precision@k, where each correctly predicted label j is associated with a weight g_j (gain). The weighted precision@k is then defined as:

$$\mathrm{wP@}k\left(\boldsymbol{y}, \widehat{\boldsymbol{y}}^{@k}\right) \coloneqq \frac{1}{k}\sum_{j=1}^{m} g_j y_j \widehat{y}_j \,. \tag{3.2}$$

Obviously, the standard variant of precision is a special case of weighted precision, where $g_j = 1$ for all j.

Let us analyze the expected value of weighted precision@k conditioned on

instance $\boldsymbol{x}$ (under the expected instance-wise utility (EIU) framework):

$$\begin{aligned}\Phi_{\mathrm{wP@}k}\left(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}\right)) &= \mathbb{E}_{\boldsymbol{y} \sim \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}]}\left[\mathrm{wP@}k\left(\boldsymbol{y}, \boldsymbol{h}^{@k}(\boldsymbol{x})\right)\right] \\ &= \sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}] \frac{1}{k} \sum_{j=1}^{m} g_j y_j \widehat{y}_j \\ &= \frac{1}{k} \sum_{j=1}^{m} g_j \widehat{y}_j \sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}] y_j \\ &= \frac{1}{k} \sum_{j=1}^{m} g_j \widehat{y}_j \eta_j(\boldsymbol{x})\,. \end{aligned} \tag{3.3}$$

From the above result, it is easy to notice that the optimal classifier $\boldsymbol{h}^{\star}_{\mathrm{wP@}k}$ for weighted precision@k predicts k labels with the highest marginal probabilities $\eta_j(\boldsymbol{x})$ multiplied by weight g_j with ties solved in any way [Wydmuch et al., 2018, 2021]:

$$\boldsymbol{h}^{\star}_{\mathrm{P@}k}(\boldsymbol{x}) \coloneqq [y_1^{\star}, y_2^{\star}, \ldots, y_m^{\star}]\,, \tag{3.4}$$

where

$$y_j^{\star}(\boldsymbol{x}) \coloneqq \mathbb{1}\left[j \in \operatorname*{arg\,top\text{-}}_{j' \in [m]} k \;\; g_{j'} \eta_{j'}(\boldsymbol{x})\right], \tag{3.5}$$

and arg top-k is a set of k arguments given with the highest values of some function (with ties broken arbitrarily).

Because we often use similar definitions of the optimal classifier throughout this work, we also define a helper function select-top-$k : \mathbb{R}^m \longrightarrow \mathcal{Y}^{@k}$ that returns a k-hot encoded vector with ones on positions corresponding to the highest values in the input vector. Using this vector, we can write the optimal classifier for precision@k as:[1]

$$\boldsymbol{h}^{\star}_{\mathrm{P@}k}(\boldsymbol{x}) \coloneqq \text{select-top-}k(\boldsymbol{g} \odot \boldsymbol{\eta}(\boldsymbol{x}))\,. \tag{3.6}$$

The weighted precision@k allows assigning different gains to different labels, for example, higher weights can be assigned to less frequent labels to increase their importance. A popular variant of such a metric is propensity-scored precision@k [Jain et al., 2016], where the gains g_j correspond to inverse propensities estimated using a data-dependent model proposed by the authors. We take a closer look at these metrics in Chapter 4.

3.2 General instance-wise weighted utilities@k

Hamming score (and its counterpart Hamming loss) is a very popular measure in classical multi-label classification. It counts the number of correct predictions, both positive and negative (or incorrect predictions in case of the loss variant) for all labels. It is less frequently used in XMLC, because of a large imbalance

[1] We use $\boldsymbol{a} \odot \boldsymbol{b} = [a_1 b_1, a_2 b_2, \ldots, a_n b_n]$ to denote the element-wise (Hadamard) product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ of size n.

in the number of positive and negative examples per instance, a naive classifier predicting all labels as negatives gets a high score (close to 1). The "at k" variant of Hamming score is defined as follows:

$$\mathrm{HS@}k\left(\boldsymbol{y}, \widehat{\boldsymbol{y}}^{@k}\right) \coloneqq \frac{1}{m}\sum_{j=1}^{m}\left(y_j\widehat{y}_j + (1-y_j)(1-\widehat{y}_j)\right) . \tag{3.7}$$

Similarly to precision@k, we can consider a weighted variant of Hamming score@k, with weights for the correct prediction of each positive and negative:

$$\mathrm{wHS@}k\left(\boldsymbol{y}, \widehat{\boldsymbol{y}}^{@k}\right) \coloneqq \frac{1}{m}\sum_{j=1}^{m}\left(g_{j,\mathrm{tp}}y_j\widehat{y}_j + g_{j,\mathrm{tn}}(1-y_j)(1-\widehat{y}_j)\right) . \tag{3.8}$$

Both precision@k, Hamming score@k, and their weighted variants are part of a more general family of instance-wise weighted utility that simply assigns utility (or cost) to each correct and wrong prediction for each label:

$$\mathrm{U@}k\left(\boldsymbol{y}, \widehat{\boldsymbol{y}}^{@k}\right) = \sum_{j=1}^{m} g_{j,\mathrm{tn}}(1-y_j)(1-\widehat{y}_j) + g_{j,\mathrm{fp}}(1-y_j)\widehat{y}_j + g_{j,\mathrm{fn}}y_j(1-\widehat{y}_j) + g_{j,\mathrm{tp}}y_j\widehat{y}_j . \tag{3.9}$$

which is defined using a gain matrix $\mathbf{G} \coloneqq [\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_m]$, where each $\boldsymbol{g}_j = [g_{\mathrm{tn}}, g_{\mathrm{fp}}, g_{\mathrm{fn}}, g_{\mathrm{tp}}]$ to express the utility of true negative, false positive, false negative, and true positive predictions respectively, for all labels $j \in [m]$. Note that $\boldsymbol{g}_j$ can be different for each label. This gives precision@k for $g_{j,\mathrm{tp}} = \frac{1}{k}, g_{j,\mathrm{tn}} = g_{j,\mathrm{fp}} = g_{j,\mathrm{fn}} = 0$ for all $j \in [m]$, and its weighted variant with $g_{j,\mathrm{tp}} = \frac{w_j}{k}$. And Hamming score is given by $g_{j,\mathrm{tp}} = g_{j,\mathrm{tn}} = \frac{1}{m}, g_{j,\mathrm{fp}} = g_{j,\mathrm{fn}} = 0$ for all $j \in [m]$.

For any $\mathbf{G}$, the optimal classifier has an appealing, simple form of returning k labels with the highest values of the affine function of marginals $\boldsymbol{\eta}(\boldsymbol{x})$, as we can suspect from our previous analysis of precision@k:

Theorem 3.2.1 (Optimal classifier of instance-wise weighted utilities)**.** *The optimal classifier* $\boldsymbol{h}^{\star}_{\mathrm{U}@k} \in \arg\max_{\boldsymbol{h}\in\mathcal{H}^{@k}} \Phi_{\mathrm{U}@k}(\boldsymbol{h})$ *for* $U(\boldsymbol{y}, \boldsymbol{h}^{@k}(\boldsymbol{x}))$ *and any* $\mathbf{G}$ *is given by:*

$$\boldsymbol{h}^{\star}_{\mathrm{U}@k}(\boldsymbol{x}) \coloneqq \text{select-top-}k(\boldsymbol{a} \odot \boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{b}) , \tag{3.10}$$

where:

$$\boldsymbol{a} = \boldsymbol{g}_{:,\mathrm{tn}} + \boldsymbol{g}_{:,\mathrm{tp}} - \boldsymbol{g}_{:,\mathrm{fp}} - \boldsymbol{g}_{:,\mathrm{fn}} , \qquad \boldsymbol{b} = \boldsymbol{g}_{:,\mathrm{fp}} - \boldsymbol{g}_{:,\mathrm{tn}} . \tag{3.11}$$

Proof. We follow a similar analysis we did for precision@k:

$$\begin{aligned}
\Phi_{\mathrm{U}@k}\left(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}\right) &= \mathbb{E}_{\boldsymbol{y} \sim \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}]}\left[U\left(\boldsymbol{y}, \boldsymbol{h}^{@k}(\boldsymbol{x}), \mathbf{G}\right)\right] \\
&= \sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}] \sum_{j=1}^{m} g_{j,\mathrm{tn}}(1 - y_j)(1 - \widehat{y}_j) \\
&\quad + g_{j,\mathrm{fp}}(1 - y_j)\widehat{y}_j + g_{j,\mathrm{fn}} y_j (1 - \widehat{y}_j) + g_{j,\mathrm{tp}} y_j \widehat{y}_j \\
&= \sum_{j=1}^{m} \left(g_{j,\mathrm{tn}}(1 - \widehat{y}_j) + g_{j,\mathrm{fp}}\widehat{y}_j\right) \left(\sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}](1 - y_j)\right) \\
&\quad + \left(g_{j,\mathrm{fn}}(1 - \widehat{y}_j + g_{j,\mathrm{tp}}\widehat{y}_j\right) \left(\sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}] y_j\right) \\
&= \sum_{j=1}^{m} g_{j,\mathrm{tn}}(1 - \widehat{y}_j)(1 - \eta_j(\boldsymbol{x})) + g_{j,\mathrm{fp}}\widehat{y}_j(1 - \eta_j(\boldsymbol{x})) \\
&\quad + g_{j,\mathrm{fn}}(1 - \widehat{y}_j)\eta_j(\boldsymbol{x}) + g_{j,\mathrm{tp}}\widehat{y}_j\eta_j(\boldsymbol{x}) \\
&= \sum_{j=1}^{m} \left(a_j \eta_j(\boldsymbol{x}) + b_j\right) \widehat{y}_j + r_j \,, \qquad (3.12)
\end{aligned}$$

where a_j, b_j are elements of the vectors calculated as in (3.11), and $r_j = g_{j,\mathrm{tn}}(1 - \eta_j(\boldsymbol{x})) + g_{j,\mathrm{fn}}$. Since r_j does not depend on the prediction of the classifier, for each $\boldsymbol{x} \in \mathcal{X}$, the objective can be maximized by the choice of $\boldsymbol{h}^{@k}(\boldsymbol{x})$ that selects k labels with the highest values of $a_j \odot \eta_j(\boldsymbol{x}) + b_j$ as defined in (3.10). □

Interestingly, this results in precision@k and Hamming score@k having the same form of the optimal classifier. As for precision@k and its weighted variant, we get the same optimal classifier as the one we derived before in (3.6), while for Hamming score $a_j = \frac{2}{m}, b_j = -\frac{1}{m}$ for all $j \in [m]$ and thus for any $\boldsymbol{x} \in \mathcal{X}$, the optimal prediction $\boldsymbol{h}^{\star}_{\mathrm{HS}}(\boldsymbol{x})$ also returns k labels with the largest marginals $\boldsymbol{\eta}_j(\boldsymbol{x})$.

In practice, $\boldsymbol{\eta}(\boldsymbol{x})$ are not available, and to apply the optimal inference procedure we need to substitute them with the estimates $\widehat{\boldsymbol{\eta}}(\boldsymbol{x})$ coming from label probability estimator (LPE). Because of that, we are interested in quantifying the regret that the classifier suffers in this case. The conditional regret for general instance-wise weighted utilities is:

$$\begin{aligned}
\mathrm{Reg}_{\mathrm{U}@k}\left(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}\right) &= \Phi_{\mathrm{U}@k}\left(\boldsymbol{h}^{@k,\star} \,|\, \boldsymbol{x}\right) - \Phi_{\mathrm{U}@k}\left(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}\right) \\
&= \sum_{j=1}^{m} \left(\left(a_j \eta_j(\boldsymbol{x}) + b_j\right) y_j^{\star} + r_j\right) - \sum_{j=1}^{m} \left(\left(a_j \eta_j(\boldsymbol{x}) + b_j\right) \widehat{y}_j + r_j\right) . \qquad (3.13)
\end{aligned}$$

The conditional regret with respect to general instance-wise weighted utilities can be upper-bounded by the L_1-estimation errors as stated by the following theorem. This theorem is a generalization of the result published in [Wydmuch et al., 2018] for precision@k.

Theorem 3.2.2. *For any distribution $\mathbb{P}[\boldsymbol{y} \mid \boldsymbol{x}]$, instance-wise weighted utility defined using gain matrix $\mathbf{G}$, estimates $\widehat{\boldsymbol{\eta}}(\boldsymbol{x})$, and a classifier $\mathcal{H}^{@k} \ni \boldsymbol{h}^{@k} =$ select-top-$k(\boldsymbol{a} \odot \widehat{\boldsymbol{\eta}}(\boldsymbol{x}) + \boldsymbol{b})$, the following holds:*

$$\operatorname{Reg}_{\mathrm{U}@k}\left(\boldsymbol{h}^{@k} \mid \boldsymbol{x}\right) \leqslant 2k \max_{j \in [m]} a_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| , \tag{3.14}$$

where $\boldsymbol{a} = \boldsymbol{g}_{:,\mathrm{tn}} + \boldsymbol{g}_{:,\mathrm{tp}} - \boldsymbol{g}_{:,\mathrm{fp}} - \boldsymbol{g}_{:,\mathrm{fn}}$, $\boldsymbol{b} = \boldsymbol{g}_{:,\mathrm{fn}} - \boldsymbol{g}_{:,\mathrm{tn}}$.

Proof. To prove this, let us add and subtract the following two terms:

$$\sum_{j=1}^{m} \left(\left(a_j \widehat{\eta}_j(\boldsymbol{x}) + b_j\right) y_j^\star + r_j\right) , \qquad \sum_{j=1}^{m} \left(\left(a_j \widehat{\eta}_j(\boldsymbol{x}) + b_j\right) \widehat{y}_j + r_j\right) , \tag{3.15}$$

from the regret and reorganize the expression in the following way:

$$\begin{aligned}
\operatorname{Reg}_{\mathrm{U}@k}(\boldsymbol{h}^{@k} \mid \boldsymbol{x}) = {} & \underbrace{\sum_{j=1}^{m} \left(\left(a_j \eta_j(\boldsymbol{x}) + b_j\right) y_j^\star + r_j\right) - \sum_{j=1}^{m} \left(\left(a_j \widehat{\eta}_j(\boldsymbol{x}) + b_j\right) y_j^\star + r_j\right)}_{\leqslant \sum_{j=1}^{m} a_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| y_j^\star} \\
& + \underbrace{\sum_{j=1}^{m} \left(\left(a_j \widehat{\eta}_j(\boldsymbol{x}) + b_j\right) \widehat{y}_j + r_j\right) - \sum_{j=1}^{m} \left(\left(a_j \eta_j(\boldsymbol{x}) + b_j\right) \widehat{y}_j + r_j\right)}_{\leqslant \sum_{j=1}^{m} a_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| \widehat{y}_j} \\
& + \underbrace{\sum_{j=1}^{m} \left(\left(a_j \widehat{\eta}_j(\boldsymbol{x}) + b_j\right) y_j^\star + r_j\right) - \sum_{j=1}^{m} \left(\left(a_j \widehat{\eta}_j(\boldsymbol{x}) + b_j\right) \widehat{y}_j + r_j\right)}_{\leqslant 0} \\
\leqslant {} & \sum_{j=1}^{m} a_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| y_j^\star + \sum_{j=1}^{m} a_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| \widehat{y}_j \, .
\end{aligned} \tag{3.16}$$

Next, we bound each L_1 error, $a_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right|$ by $\max_{j \in [m]} a_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right|$. There are at most $\|\boldsymbol{y}^{@k,\star} \vee \widehat{\boldsymbol{y}}^{@k}\|_1 \leqslant 2k$ such terms that stay positive. Therefore:

$$\operatorname{Reg}_{\mathrm{U}@k}\left(\boldsymbol{h}^{@k} \mid \boldsymbol{x}\right) \leqslant 2k \max_{j \in [m]} a_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| . \tag{3.17}$$

□

We can see that the bound does not depend on m. However, if $k = m$, then only one prediction vector is possible, resulting in $\boldsymbol{y}^{@k,\star} = \widehat{\boldsymbol{y}}^{@k}$ and $\operatorname{Reg}_{\mathrm{U}@k} = 0$. If $m < 2k$, then $\|\widehat{\boldsymbol{y}}^{@k,\star} \vee \widehat{\boldsymbol{y}}^{@k}\|_1 \leqslant m$, as some of the positive labels from $\boldsymbol{y}^{@k,\star}$ have to be positive also in $\widehat{\boldsymbol{y}}^{@k}$, so the bound can be tighter in this specific case. However, we do not consider it further, as in extreme classification $k \ll m$. For standard precision@k, $a_j = \frac{1}{k}$ that cancels k in the bound, making it independent of k and resulting in the original bound obtained by Wydmuch et al. [2018].

3.3 Recall@k

Another very popular instance-wise metric is recall@k, which measures the fraction of correct predictions among all positive labels. It is defined as follows:

$$\mathrm{R@}k\left(\boldsymbol{y}, \widehat{\boldsymbol{y}}^{@k}\right) = \frac{1}{\|\boldsymbol{y}\|_1} \sum_{j=1}^{m} y_j \widehat{y}_j \,. \tag{3.18}$$

While it seems to be similar to precision@k, it does not belong to the general family of general instance-wise weighted utilities, and the optimal classifier for recall@k is different. To show that, let us analyze the expected value of recall@k in a similar way as we did before:

$$\begin{aligned}
\Phi_{\mathrm{R@}k}(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}) &= \mathbb{E}_{\boldsymbol{y} \sim \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}]} \left[\mathrm{R@}k \left(\boldsymbol{y}, \boldsymbol{h}^{@k}(\boldsymbol{x}) \right) \right] \\
&= \sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}] \frac{1}{\|\boldsymbol{y}\|_1} \sum_{j=1}^{m} y_j \widehat{y}_j \\
&= \sum_{j=1}^{m} \widehat{y}_j \sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}] \frac{y_j}{\|\boldsymbol{y}\|_1} \\
&= \sum_{j=1}^{m} \widehat{y}_j \eta'_j(\boldsymbol{x}) \,,
\end{aligned} \tag{3.19}$$

where

$$\eta'_j(\boldsymbol{x}) \coloneqq \sum_{\boldsymbol{y} \in \mathcal{Y}} \frac{y_j}{\|\boldsymbol{y}\|_1} \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}] \,, j \in \boldsymbol{y} \,. \tag{3.20}$$

The optimal classifier $\boldsymbol{h}^{\star}_{\mathrm{R@}k}$ for recall@k predicts k labels with the highest values of $\eta'_j(\boldsymbol{x})$ with ties solved in any way [Menon et al., 2019]:

$$\boldsymbol{h}^{\star}_{\mathrm{R@}k}(\boldsymbol{x}) \coloneqq \text{select-top-}k(\boldsymbol{\eta}'(\boldsymbol{x})) \,. \tag{3.21}$$

Unfortunately, in the general case the classifier $\boldsymbol{h}^{\star}_{\mathrm{R@}k}$ is not optimal for precision@k and vice versa [Wydmuch et al., 2018, Menon et al., 2019], as $\eta'_j(\boldsymbol{x})$ differ from marginal conditional probabilities $\eta_j(\boldsymbol{x})$. The situation changes when the labels are conditionally independent, that is, if for each $\boldsymbol{y} \in \mathcal{Y}$:

$$\mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}] = \prod_{j=1}^{m} \mathbb{P}[y_j \,|\, \boldsymbol{x}] \,. \tag{3.22}$$

Theorem 3.3.1. *Given conditionally independent labels, $\eta_j(\boldsymbol{x})$ and $\eta'_j(\boldsymbol{x})$, $j \in [m]$ induce the same order of labels.*

Proof (sketch, full proof in Appendix A.1.1). It is enough to show that, in the case of conditionally independent labels, sorting according to $\boldsymbol{\eta}'(\boldsymbol{x})$ is equivalent to sorting according to $\boldsymbol{\eta}(\boldsymbol{x})$. Let labels i and j be so that $\eta_i(\boldsymbol{x}) \geqslant \eta_j(\boldsymbol{x})$. Then, when summing over $\boldsymbol{y} \in \mathcal{Y} : y_j = 1$ in (3.20), we are interested in four different

subsets of $\mathcal{Y}$, $S_{i,j}^{u,w} = \{\boldsymbol{y} \in \mathcal{Y} : y_i = u \wedge y_j = w\}$, where $u, w \in \{0, 1\}$. Remark that during mapping none of $\boldsymbol{y} \in S_{i,j}^{0,0}$ plays any role, and for each $\boldsymbol{y} \in S_{i,j}^{1,1}$, the value of $\frac{y_t \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}]}{\sum_{t'=1}^{m} y_{t'}}$, for $t \in \{i, j\}$, is the same for both y_i and y_j. Now, let $\boldsymbol{y}' \in S_{i,j}^{1,0}$ and $\boldsymbol{y}'' \in S_{i,j}^{0,1}$ be the same on all elements except the i-th and the j-th one. Then, because of the label independence and the assumption that $\eta_i(\boldsymbol{x}) \geqslant \eta_j(\boldsymbol{x})$, we have $\mathbb{P}[\boldsymbol{y}' \,|\, \boldsymbol{x}] \geqslant \mathbb{P}[\boldsymbol{y}'' \,|\, \boldsymbol{x}]$. Therefore, after mapping (3.20) we obtain $\eta'_i(\boldsymbol{x}) \geqslant \eta'_j(\boldsymbol{x})$. □

As in the case of general instance-wise weighted utilities at k, we can provide a regret bound of an optimal classifier that uses estimates of $\boldsymbol{\eta}'(\boldsymbol{x})$ coming from a label probability estimator (LPE):

Theorem 3.3.2. *For any distribution* $\mathbb{P}(\boldsymbol{y} \,|\, \boldsymbol{x})$ *and the classifier* $\mathcal{H}^{@k} \ni \boldsymbol{h}^{@k} =$ select-top-$k(\widehat{\boldsymbol{\eta}}'(\boldsymbol{x}))$*, the following holds:*

$$\operatorname{Reg}_{\mathrm{R}@k}\left(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}\right) \leqslant 2k \max_{j \in [m]} \left|\eta'_j(\boldsymbol{x}) - \widehat{\eta}'_j(\boldsymbol{x})\right| . \tag{3.23}$$

The proof follows exactly the same line of reasoning as the proof of Theorem 3.2.2 using the form of $\Phi_{\mathrm{R}@k}(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x})$ derived in (3.19). We present it in Appendix A.1.2.

Pick-one-label heuristic and other reductions

Interestingly, the $\boldsymbol{\eta}'(\boldsymbol{x})$ values can be estimated using the pick-one-label heuristic, which is sometimes used to transform a multi-label classification problem to a multi-class classification problem [Joulin et al., 2017, Jernite et al., 2017]. This heuristic randomly (using the uniform distribution) picks one of the positive labels from a given training observation as the only one positive. The resulting observation is then treated as a multi-class observation. Since the probability of picking a positive label j is equal to $y_j / \|\boldsymbol{y}\|_1$, the pick-one-label heuristic maps the multi-label distribution to a multi-class distribution in the following way:

$$\eta'_j(\boldsymbol{x}) \coloneqq \mathbb{P}'[y_j = 1 \,|\, \boldsymbol{x}] = \sum_{\boldsymbol{y} \in \mathcal{Y}} \frac{y_j}{\|\boldsymbol{y}\|_1} \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}] \,, j \in \boldsymbol{y} \,. \tag{3.24}$$

The resulting $\eta'_j(\boldsymbol{x})$ forms a multi-class distribution as the probabilities sum up to 1.There also exist other transformations of the multi-label problem that result in the estimation of $\eta'_j(\boldsymbol{x})$. Moreover, Menon et al. [2019] shows that a whole family of related reductions—most notably the normalized one-versus-all (OVA-N) and normalized pick-all-labels (PAL-N) methods implicitly target the exact same transformed marginals $\eta'_j(\boldsymbol{x})$ (3.24), being consistent for recall@k, while unnormalized one-versus-all (OVA) and pick-all-labels (PAL) reductions are consistent with respect to precision@k.

3.4 DCG@k and nDCG@k

The last metrics we discuss in this section are discounted cumulative gain at k (DCG@k) and its normalized variant (nDCG@k). These metrics, adapted from Information Retrieval [Järvelin and Kekäläinen, 2002] similarly to precision@k and recall@k, accumulate the number of correctly predicted labels among the k predicted labels, however, they differ from the other performance metrics we have considered so far, as they take not only k predicted labels into account, but also the order among them. The DCG@k and nDCG@k additionally assign different discounts to the correctly predicted labels, depending on their order, that needs to be additionally specified by the classifier. The motivation behind it is that in many applications, the predictions are displayed in a specific order to the user, and predicting relevant labels in the first position is more rewarding than predicting them in the latter position, as they will be noticed first. The DCG@k is defined as follows:

$$\mathrm{DCG@}k\left(\boldsymbol{y}, \boldsymbol{h}^{@k}(\boldsymbol{x})\right) \coloneqq \sum_{j=1}^{m} y_j \widehat{y}_j d_{\mathrm{rank}(j)}\,, \tag{3.25}$$

where $\mathrm{rank}(j)$ is a position of the label j in the order predicted by classifier $\boldsymbol{h}^{@k}$, notice that it is enough to define it for top k predicted labels. Based on the position $\mathrm{rank}(j)$, a different discount factor $d_{\mathrm{rank}(j)}$ is assigned to the label j. The $\boldsymbol{d} \in \mathbb{R}^m$ denotes all the discount factors for all possible positions. For DCG@k, we assume that its elements d_i should be non-increasing with i. If all elements of $\boldsymbol{d}$ are equal to 1, DCG@k reduces to precision@k. The most popular form of discount factors is:

$$d_i \coloneqq \frac{1}{\log_2(i+1)}\,, \tag{3.26}$$

Because the DCG@k is a function with a counter-domain that may not fit into the $[0, 1]$ range, it is common to use its normalized variant nDCG@k, that normalizes the DCG@k by the ideal (highest possible value) DCG@k for a given label vector $\boldsymbol{y}$, denoted as iDCG@k. It is defined as follows:

$$\mathrm{iDCG@}k \coloneqq \sum_{i=1}^{k} d_i\,, \tag{3.27}$$

resulting in the following definition of nDCG@k:

$$\mathrm{nDCG@}k\left(\boldsymbol{y}, \boldsymbol{h}^{@k}(\boldsymbol{x})\right) \coloneqq \frac{\mathrm{DCG@}k\left(\boldsymbol{y}, \boldsymbol{h}^{@k}(\boldsymbol{x})\right)}{\mathrm{iDCG@}k}\,. \tag{3.28}$$

Similarly to precision@k and Hamming Score, we can also consider weighted variants of DCG@k and nDCG@k, that assign additional weight g_j (gain) that

depends on label j:

$$\begin{aligned}\mathrm{wDCG@}k(\boldsymbol{y}, \boldsymbol{h}^{@k}(\boldsymbol{x})) &\coloneqq \sum_{j=1}^{m} g_j y_j \widehat{y}_j d_{\mathrm{rank}(j)}\,, \\ \mathrm{wnDCG@}k(\boldsymbol{y}, \boldsymbol{h}^{@k}(\boldsymbol{x})) &\coloneqq \frac{\mathrm{wDCG@}k(\boldsymbol{y}, \boldsymbol{h}^{@k}(\boldsymbol{x}))}{\mathrm{iDCG@}k}\,. \end{aligned} \tag{3.29}$$

As in the case of precision, popular weighted variants of these metrics are propensity-scored DCG@k and propensity-scored nDCG@k [Jain et al., 2016]. As in the case of propensity-scored precision@k, the weights g_j correspond to inverse propensities.

Once again we start, we analyze the expected value of wDCG@k given an instance $\boldsymbol{x}$:

$$\begin{aligned}\Phi_{\mathrm{wDCG@}k}\left(\boldsymbol{h}^{@k} \mid \boldsymbol{x}\right) &= \mathbb{E}_{\boldsymbol{y} \sim \mathbb{P}[\boldsymbol{y} \mid \boldsymbol{x}]}\left[\mathrm{wDCG@}k\left(\boldsymbol{y}, \boldsymbol{h}^{@k}(\boldsymbol{x})\right)\right] \\ &= \sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{P}[\boldsymbol{y} \mid \boldsymbol{x}] \sum_{j=1}^{m} g_j y_j \widehat{y}_j d_{\mathrm{rank}(j)} \\ &= \sum_{j=1}^{m} d_{\mathrm{rank}(j)} g_j \widehat{y}_j \sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{P}[\boldsymbol{y} \mid \boldsymbol{x}] y_j \\ &= \sum_{j=1}^{m} d_{\mathrm{rank}(j)} g_j \widehat{y}_j \eta_j(\boldsymbol{x})\,, \end{aligned} \tag{3.30}$$

from the above result, we can easily notice that the optimal classifier $\boldsymbol{h}^{\star}_{\mathrm{wDCG@}k}$ for wDCG@k is to select k labels with the highest values of $d_{\mathrm{rank}(j)} g_j \eta_j(\boldsymbol{x})$. Since $d_{\mathrm{rank}(j)}$ depends on the predicted rank of the label j and d is decreasing with the rank, we conclude that the prediction should be k labels with the highest values of $g_j \eta_j(\boldsymbol{x})$, same as in the case of weighted precision@k, but additionally sorted in descending order of $g_j \eta_j(\boldsymbol{x})$:

$$\begin{aligned}\boldsymbol{h}^{\star}_{\mathrm{wDCG@}k}(\boldsymbol{x}) &\coloneqq \text{select-top-}k(\boldsymbol{g} \odot \boldsymbol{\eta}(\boldsymbol{x}))\,, \\ \mathrm{rank}^{\star}_{\mathrm{wDCG@}k}(j) &\coloneqq \mathrm{argsort}_{\downarrow}(\boldsymbol{g} \odot \boldsymbol{\eta}(\boldsymbol{x}))_j\,, \end{aligned} \tag{3.31}$$

where $\mathrm{argsort}_{\downarrow} : \mathbb{R}^m \longrightarrow \mathbb{N}^m$ returns a vector of indices that would sort the input vector according to descending order. Since iDCG@k is a constant factor, the optimal classifier for wnDCG@k is the same as for wDCG@k.

Again, we can also provide a regret bound of an optimal classifier that uses estimates of $\boldsymbol{\eta}(\boldsymbol{x})$ coming from a label probability estimator (LPE):

Theorem 3.4.1. *For any distribution $\mathbb{P}(\boldsymbol{y} \mid \boldsymbol{x})$, vector of label gains $\boldsymbol{g}$, vector of discount factors $\boldsymbol{d}$, and the classifier $\mathcal{H}^{@k} \ni \boldsymbol{h}^{@k} = \text{select-top-}k(g \odot \widehat{\boldsymbol{\eta}}(\boldsymbol{x}))$, the*

following holds:

$$\begin{aligned}\operatorname{Reg}_{\text{wDCG@}k}\left(\boldsymbol{h}^{@k} \mid \boldsymbol{x}\right) &\leqslant 2\sum_{i}^{k} d_i \max_{j\in[m]} g_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| ,\\ \operatorname{Reg}_{\text{wnDCG@}k}\left(\boldsymbol{h}^{@k} \mid \boldsymbol{x}\right) &\leqslant 2\max_{j\in[m]} g_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| .\end{aligned} \tag{3.32}$$

Proof (sketch, as it similar to the previous proofs, full proof in Appendix A.1.3). Following a similar line of reasoning as the proof of Theorem 3.2.2, we show that:

$$\begin{aligned}\operatorname{Reg}_{\text{wDCG@}k}\left(\boldsymbol{h}^{@k} \mid \boldsymbol{x}\right) \leqslant &\sum_{j=1}^{m} d_{\operatorname{rank}(\boldsymbol{\eta}(\boldsymbol{x}),j)} g_j |\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})| y_j^{\star}\\ &+ \sum_{j=1}^{m} d_{\operatorname{rank}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}),j)} g_j |\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})| \widehat{y}_j .\end{aligned} \tag{3.33}$$

Once again, we can bound $g_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right|$ by $\max_{j\in[m]} g_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right|$. Notice that both $\boldsymbol{y}^{@k,\star}$ and $\widehat{\boldsymbol{y}}^{@k}$ contain exactly k positive labels and each label appears at only one, unique rank. Therefore $\sum_{j=1}^{m} d_{\operatorname{rank}(\boldsymbol{\eta}(\boldsymbol{x}),j)} y_j^{\star} = \sum_{j=1}^{m} d_{\operatorname{rank}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}),j)} \widehat{y}_j = \sum_{i=1}^{k} d_i$ and we end up with the final bound for DCG@k:

$$\operatorname{Reg}_{\text{wDCG@}k}\left(\boldsymbol{h}^{@k} \mid \boldsymbol{x}\right) \leqslant 2\sum_{i}^{k} d_i \max_{j\in[m]} g_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| . \tag{3.34}$$

nDCG@k is DCG@k divided by constant iDCG@k (3.27) that cancels $\sum_{i}^{k} d_i$ term resulting in the final bound:

$$\operatorname{Reg}_{\text{wnDCG@}k}\left(\boldsymbol{h}^{@k} \mid \boldsymbol{x}\right) \leqslant 2\max_{j\in[m]} g_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| . \tag{3.35}$$

□

Interestingly, the bound for weighted nDCG@k is the same as for weighted precision@k.

Alternative formulation of nDCG

In the literature, we can find an alternative formulation of nDCG@k with less naive normalization, that takes into account that for a given sample, the number of positive labels might be lower than k (i.e., $\|\boldsymbol{y}\|_1 < k$). It then defines iDCG@k differently:

$$\text{iDCG@}k'(\boldsymbol{y}) = \sum_{i=1}^{\min(k,\|\boldsymbol{y}\|_1)} g_i . \tag{3.36}$$

While this seems to be a minor change, it has a significant impact on the optimal classifier for nDCG@k. The expected value of nDCG@k given an

instance $\boldsymbol{x}$ is then:

$$\begin{aligned}\Phi_{\text{nDCG@}k'}\left(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}\right) &= \mathbb{E}_{\boldsymbol{y} \sim \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}]}\left[\text{nDCG@}k'\left(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})\right)\right] \\ &= \sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}] \frac{\sum_{j=1}^{m} y_j \widehat{y}_j d_{\text{rank}(j)}}{\text{iDCG@}k'(\boldsymbol{y})} \\ &= \sum_{j=1}^{m} d_{\text{rank}(j)} \widehat{y}_j \sum_{\boldsymbol{y} \in \mathcal{Y}} \frac{\mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}] y_j}{\text{iDCG@}k'(\boldsymbol{y})} \,. \end{aligned} \tag{3.37}$$

Now with iDCG@k depending on $\boldsymbol{y}$, we cannot easily express the optimal classifier for nDCG@k as a function of marginal conditional probabilities $\eta_j(\boldsymbol{x})$ as we did before. Only for $k = 1$, when iDCG@k is equal to g_1 or 0, the optimal classifier for nDCG@k is to predict the label with the highest value of $\eta_j(\boldsymbol{x})$. For $k > 1$, in general, the optimal classifiers for nDCG@k and DCG@k are different, except in the case when the labels are conditionally independent as in (3.22).

Theorem 3.4.2. *Given conditionally independent labels, $\eta_j(\boldsymbol{x})$ and $\sum_{\boldsymbol{y} \in \mathcal{Y}} \frac{\mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}] y_j}{\text{iDCG@}k'(\boldsymbol{y})}$, $j \in [m]$, induce the same order of labels.*

The proof and further discussion on that formulation of nDCG@k can be found in [Jasinska and Dembczyński, 2018].

3.5 Summary of the chapter

In this chapter, we have shown the form of the optimal (Bayes) classifier for popular instance-wise metrics used in extreme classification, that is, for:

1. the family of instance-wise weighted utilities, which includes precision@k, Hamming score@k, and their weighted variants such as the popular propensity-scored precision@k,
2. recall@k,
3. DCG@k and nDCG@k, as well as their weighted variants that include propensity-scored DCG@k and nDCG@k.

All the Bayes classifiers, except the one for recall@k, are determined by the marginal conditional probabilities of labels, $\eta_j(\boldsymbol{x}) = \mathbb{P}[y_j = 1 \,|\, \boldsymbol{x}]$. This implies that estimating the conditional probability of labels and then predicting the labels according to the form of the optimal classifier is a consistent strategy for predicting under these metrics. We have additionally presented regret bounds for those strategies when using estimates of $\eta_j(\boldsymbol{x})$ coming from a label probability estimator (LPE), which suffer L_1 estimation error.

4

Instance-wise metrics at k under the missing labels setting

In this chapter, we discuss the instance-wise metrics under the missing labels setting that we introduced in Section 2.3.3 and their connection to long-tail performance. We first derived unbiased variants of these metrics, whose special cases are called propensity-scored metrics and were initially proposed by Jain et al. [2016]. This work has been highly influential, and the propensity-scored metrics have become default performance metrics for problems with tail labels [You et al., 2019a, Guo et al., 2019, Babbar and Schölkopf, 2019]. We follow with the investigation of the rationale for the usage of those metrics for the evaluation of long-tail performance, based on the analysis conducted in [Schultheis et al., 2022].

4.1 Unbiased general instance-wise weighted metrics at k

We start by deriving a general form of unbiased utilities in multi-label classification under the assumption of missing labels for the expected instance-wise utility (EIU) framework. Let us recall the idea behind the unbiased utilities introduced in (2.28):

$$\forall \boldsymbol{h}: \ \Phi_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h}) = \breve{\Phi}_{\mathbb{P}[\boldsymbol{x},\breve{\boldsymbol{y}}]}(\boldsymbol{h})\,.$$

Under the EIU framework, we obtain the following definition:

Definition 4.1.1 (Unbiased utility under EIU framework)**.** For every classifier $\boldsymbol{h}$, true $\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]$ and observed distribution $\mathbb{P}[\boldsymbol{x},\breve{\boldsymbol{y}}]$ related by assumption of missing labels, a utility $\breve{\Phi}$ is said to be unbiased estimate of Φ if

$$\begin{aligned}\Phi_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h}) &= \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}[\Psi(\boldsymbol{y},\boldsymbol{h}(\boldsymbol{x}))] \\ &= \mathbb{E}_{\boldsymbol{x},\breve{\boldsymbol{y}}\sim\mathbb{P}[\boldsymbol{x},\breve{\boldsymbol{y}}]}[\breve{\Psi}(\breve{\boldsymbol{y}},\boldsymbol{h}(\boldsymbol{x}))] = \breve{\Phi}_{\mathbb{P}[\boldsymbol{x},\breve{\boldsymbol{y}}]}(\boldsymbol{h})\,. \end{aligned} \tag{4.1}$$

By assuming a label-wise model of missing labels with propensities $p_j(\boldsymbol{x}) \coloneqq \mathbb{P}\left[\breve{y}_j = 1 \,|\, y_j = 1, \boldsymbol{x}\right]$ (as defined in (2.25)), we can derive an unbiased utility for

a family of general instance-wise weighted metrics introduced in the previous chapter (3.9):

$$\mathrm{U}@k\left(\boldsymbol{y}, \widehat{\boldsymbol{y}}^{@k}\right) = \sum_{j=1}^{m} g_{j,\mathrm{tn}}(1-y_j)(1-\widehat{y}_j) + g_{j,\mathrm{fp}}(1-y_j)\widehat{y}_j + g_{j,\mathrm{fn}}y_j(1-\widehat{y}_j) + g_{j,\mathrm{tp}}y_j\widehat{y}_j\,.$$

Theorem 4.1.2 (Unbiased general instance-wise weighted utility under EIU framework)**.** *Under Definition 4.1.1 and missing labels model*

$$\begin{aligned}\breve{\mathrm{U}}@k\left(\boldsymbol{x}, \breve{\boldsymbol{y}}, \widehat{\boldsymbol{y}}^{@k}\right) = \sum_{j=1}^{m} & g_{j,\mathrm{tn}}\left(1 - \frac{\breve{y}_j}{p_j(\boldsymbol{x})}\right)(1-\widehat{y}_j) + g_{j,\mathrm{fp}}\left(1 - \frac{\breve{y}_j}{p_j(\boldsymbol{x})}\right)\widehat{y}_j \\ & + g_{j,\mathrm{fn}}\frac{\breve{y}_j}{p_j(\boldsymbol{x})}(1-\widehat{y}_j) + g_{j,\mathrm{tp}}\frac{\breve{y}_j}{p_j(\boldsymbol{x})}\widehat{y}_j\,, \end{aligned} \tag{4.2}$$

is an unbiased estimate of general instance-wise weighted utility (3.9)*.*

Proof. Let us consider a general family of instance-wise utilities that additionally decompose over labels:

$$\mathrm{U}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = \sum_{j=1}^{m} u^j(y_j, \widehat{y}_j) \tag{4.3}$$

where $\mathrm{u}^j(y_j, \widehat{y}_j)$ is a label-wise part of the utility. This covers general instance-wise weighted utilities (3.9). Let $\mathcal{S}$ be $\{0,1\}^{m-1}$ and $\boldsymbol{s}^j = [y_1, \ldots, y_{j-1}, y_{j+1}, \ldots, y_m] \in \mathcal{S}$ be a label vector without label j. Then, we have:

$$\begin{aligned}\Phi_{\mathbb{P}[\boldsymbol{y}\,|\,\boldsymbol{x}]}(\boldsymbol{h}|\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{y}\sim\mathbb{P}[\boldsymbol{y}\,|\,\boldsymbol{x}]}\left[\mathrm{U}\left(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})\right)\right] &= \sum_{\boldsymbol{y}\in\mathcal{Y}} \mathbb{P}[\boldsymbol{y}\,|\,\boldsymbol{x}] \sum_{j=1}^{m} u^j(y_j, \widehat{y}_j) \\ &= \sum_{j=1}^{m}\sum_{\boldsymbol{y}\in\mathcal{Y}} \mathbb{P}[\boldsymbol{s}^j\,|\,y_j, \boldsymbol{x}]\mathbb{P}[y_j\,|\,\boldsymbol{x}]u^j(y_j, \widehat{y}_j) \\ &= \sum_{j=1}^{m}\sum_{y_j\in\{0,1\}} \mathbb{P}[y_j\,|\,\boldsymbol{x}]u^j(y_j, \widehat{y}_j) \sum_{\boldsymbol{s}^j\in\mathcal{S}} \mathbb{P}[\boldsymbol{s}^j\,|\,y_j, \boldsymbol{x}] \\ &= \sum_{j=1}^{m}\sum_{y_j\in\{0,1\}} \mathbb{P}[y_j\,|\,\boldsymbol{x}]u^j(y_j, \widehat{y}_j) \end{aligned} \tag{4.4}$$

If the label-wise part of the metric $u^j(y_j, \widehat{y}_j)$ can be characterized by its positive $u^{j,(1)}(\widehat{y}_j)$ and negative part $u^{j,(0)}(\widehat{y}_j)$, then we have:

$$\begin{aligned}u^j(y_j, \widehat{y}_j) &= y_j u^{j,(1)}(\widehat{y}_j) + (1-y_j)u^{j,(0)}(\widehat{y}_j) \\ &= y_j\left(u^{j,(1)}(\widehat{y}_j) - u^{j,(0)}(\widehat{y}_j)\right) + u^{j,(0)}(\widehat{y}_j)\,. \end{aligned} \tag{4.5}$$

This also applies to general instance-wise weighted utilities. Combining it with assumption $\mathbb{P}[\breve{y}_j = 1 | y_j = 0, \boldsymbol{x}] = 0$ under the missing-labels setting, we can

additionally simplify the result of (4.4) to

$$\begin{aligned}\Phi_{\mathbb{P}[\boldsymbol{y}\,|\,\boldsymbol{x}]}(\boldsymbol{h}|\boldsymbol{x}) &= \sum_{j=1}^{m}\sum_{\breve{y}_j\in\{0,1\}}\mathbb{P}[y_j\,|\,\boldsymbol{x}]u^j(\breve{y}_j,\widehat{y}_j)\\ &= \sum_{j=1}^{m}\mathbb{P}[y_j=1\,|\,\boldsymbol{x}]\left(u^{j,(1)}(\widehat{y}_j)-u^{j,(0)}(\widehat{y}_j)\right)+u^{j,(0)}(\widehat{y}_j)\\ &= \sum_{j=1}^{m}\eta_j(\boldsymbol{x})\left(u^{j,(1)}(\widehat{y}_j)-u^{j,(0)}(\widehat{y}_j)\right)+u^{j,(0)}(\widehat{y}_j) \qquad (4.6)\end{aligned}$$

Now let us recall from Section 2.3.3 that $\breve{\eta}_j(\boldsymbol{x}) \coloneqq \mathbb{P}\left[\breve{y}_j=1\,|\,\boldsymbol{x}\right]$ and $\eta_j(\boldsymbol{x}) = \breve{\eta}_j(\boldsymbol{x})/p_j(\boldsymbol{x})$ (2.26), by applying it we get:

$$\begin{aligned}\Phi_{\mathbb{P}[\boldsymbol{y}\,|\,\boldsymbol{x}]}(\boldsymbol{h}|\boldsymbol{x}) &= \sum_{j=1}^{m}\frac{\breve{\eta}_j(\boldsymbol{x})}{p_j(\boldsymbol{x})}\left(u^{j,(1)}(\widehat{y}_j)-u^{j,(0)}(\widehat{y}_j)\right)+u^{j,(0)}(\widehat{y}_j)\\ &= \sum_{j=1}^{m}\frac{\mathbb{P}[\breve{y}_j=1|\boldsymbol{x}]}{p_j(\boldsymbol{x})}\left(u^{j,(1)}(\widehat{y}_j)-u^{j,(0)}(\widehat{y}_j)\right)+u^{j,(0)}(\widehat{y}_j)\\ &= \mathbb{E}_{\breve{\boldsymbol{y}}\sim\mathbb{P}[\breve{\boldsymbol{y}}\,|\,\boldsymbol{x}]}\left[\sum_{j=1}^{m}\frac{\breve{y}_j}{p_j(\boldsymbol{x})}\left(u^{j,(1)}(\widehat{y}_j)-u^{j,(0)}(\widehat{y}_j)\right)+u^{j,(0)}(\widehat{y}_j)\right]\\ &= \mathbb{E}_{\breve{\boldsymbol{y}}\sim\mathbb{P}[\breve{\boldsymbol{y}}\,|\,\boldsymbol{x}]}\left[\breve{\mathrm{U}}\left(\boldsymbol{x},\breve{\boldsymbol{y}},\boldsymbol{h}(\boldsymbol{x})\right)\right]=\breve{\Phi}_{\mathbb{P}[\breve{\boldsymbol{y}}\,|\,\boldsymbol{x}]}(\boldsymbol{h}|\boldsymbol{x})\,. \qquad (4.7)\end{aligned}$$

Resulting in the unbiased utility of the form:

$$\begin{aligned}\breve{\mathrm{U}}\left(\boldsymbol{x},\breve{\boldsymbol{y}},\widehat{\boldsymbol{y}}\right) &= \sum_{j=1}^{m}\frac{\breve{y}_j}{p_j(\boldsymbol{x})}\left(u^{j,(1)}(\widehat{y}_j)-u^{j,(0)}(\widehat{y}_j)\right)+u^{j,(0)}(\widehat{y}_j)\\ &= \sum_{j=1}^{m}\frac{\breve{y}_j}{p_j(\boldsymbol{x})}u^{j,(1)}(\widehat{y}_j)+\left(1-\frac{\breve{y}_j}{p_j(\boldsymbol{x})}\right)u^{j,(0)}(\widehat{y}_j)\,. \qquad (4.8)\end{aligned}$$

Now, by applying the positive and negative parts of the general instance-wise weighted utility, i.e.,

$$\begin{aligned}u^{j,(0)}(\widehat{y}_j) &= g_{j,\mathrm{tn}}(1-\widehat{y}_j)+g_{j,\mathrm{fp}}\widehat{y}_j\,,\\ u^{j,(1)}(\widehat{y}_j) &= g_{j,\mathrm{fn}}(1-\widehat{y}_j)+g_{j,\mathrm{tp}}\widehat{y}_j\,, \qquad (4.9)\end{aligned}$$

to (4.8), we get (4.2) which concludes the proof. □

Since precision@k is the variant of general instance-wise weighted metrics with $g_{j,\mathrm{tp}}=\frac{1}{k}$ and $g_{j,\mathrm{tn}}=g_{j,\mathrm{fp}}=g_{j,\mathrm{fn}}=0$, its unbiased variant is given by:

$$\breve{\mathrm{P}}@k\left(\boldsymbol{x},\breve{\boldsymbol{y}},\widehat{\boldsymbol{y}}^{@k}\right)\coloneqq\frac{1}{k}\sum_{j=1}^{m}\frac{\breve{y}_j\widehat{y}_j}{p_j(\boldsymbol{x})}\,. \qquad (4.10)$$

Applying similar steps as in the proof of Theorem 4.1.2, one can derive an unbiased

version of (n)DCG@k (we present the proof in Appendix A.2.1):

$$\widetilde{\mathrm{DCG}}@k\left(\boldsymbol{x}, \breve{\boldsymbol{y}}, \boldsymbol{h}^{@k}(\boldsymbol{x})\right) \coloneqq \sum_{j=1}^{m} \frac{\breve{y}_j \widehat{y}_j d_{\mathrm{rank}(j)}}{p_j(\boldsymbol{x})}\,,$$
$$\widetilde{\mathrm{nDCG}}@k\left(\boldsymbol{x}, \breve{\boldsymbol{y}}, \boldsymbol{h}^{@k}(\boldsymbol{x})\right) \coloneqq \frac{\widetilde{\mathrm{DCG}}@k\left(\boldsymbol{x}, \breve{\boldsymbol{y}}, \boldsymbol{h}^{@k}(\boldsymbol{x})\right)}{\mathrm{iDCG}@k}\,. \tag{4.11}$$

By assuming an even simpler propensity model, where propensity depends only on a label, $p_j \coloneqq \mathbb{P}\left[\breve{y}_j = 1 \,|\, y_j = 1\right]$ (as defined in (2.27)), we get the so-called propensity-scored precision@k:

$$\mathrm{PSP}@k\left(\breve{\boldsymbol{y}}, \widehat{\boldsymbol{y}}^{@k}\right) \coloneqq \frac{1}{k} \sum_{j=1}^{m} \frac{\breve{y}_j \widehat{y}_j}{p_j}\,, \tag{4.12}$$

and propensity-scored (n)DCG@k:

$$\widetilde{\mathrm{PSDCG}}@k\left(\breve{\boldsymbol{y}}, \boldsymbol{h}^{@k}(\boldsymbol{x})\right) \coloneqq \frac{1}{k} \sum_{j=1}^{m} \frac{\breve{y}_j \widehat{y}_j d_{\mathrm{rank}(j)}}{p_j}\,. \tag{4.13}$$

Some more examples of metrics using this simple propensity model can be found in [Jain et al., 2016, Table 1]. However, only the two above are commonly used, as the others cannot be fully expressed by label-wise propensities only [Bhatia et al., 2016].

Notice that both unbiased and propensity-scored versions of precision@k and (n)DCG@k are also special cases of the general weighted variants of these metrics considered in Chapter 3, where the per label weights are equal to the inverse of label propensity. This means that the optimal classifier for these metrics under missing labels remains the same:

$$\boldsymbol{h}^{\star}_{\mathrm{PSP}@k}(\boldsymbol{x}) = \boldsymbol{h}^{\star}_{\mathrm{PS(n)DCG}@k}(\boldsymbol{x}) \coloneqq \text{select-top-}k(\boldsymbol{g} \odot \boldsymbol{\eta}(\boldsymbol{x}))\,, \tag{4.14}$$

where $\boldsymbol{g} = \left[\frac{1}{p_1(\boldsymbol{x})}, \ldots, \frac{1}{p_m(\boldsymbol{x})}\right]$.

Unbiased surrogate losses

It is worth noticing that the above proof of Theorem 4.1.2 can also be applied to many popular surrogate losses, as they are naturally instance-wise and decompose into positive and negative parts. This way, one can obtain, for example, an unbiased variant of logistic loss [Saito et al., 2020, Qaraei et al., 2021]:

$$\breve{\ell}_{\log}(\boldsymbol{x}, \breve{y}_j, \widehat{y}_j) = -\frac{\breve{y}_j}{p_j(\boldsymbol{x})} \log(\widehat{y}_j) - \left(1 - \frac{\breve{y}_j}{p_j(\boldsymbol{x})}\right) \log(1 - \widehat{y}_j)\,. \tag{4.15}$$

By minimizing the above loss on observed labels, one can obtain an estimator of true marginal conditional probabilities $\eta_j(\boldsymbol{x})$.

Of course, to use one of such unbiased utilities (or surrogate losses), one needs to know a propensity in advance, either $p_j(\boldsymbol{x})$ or their simpler variant p_j. Unfortunately, this might be difficult in practice.

In some cases, it might be possible to collect labels under controlled bias, which allows for easy calculation of propensities, e.g., by verifying a label of an instance selected randomly using a uniform distribution. This is a standard approach used in recommendation systems [Saito et al., 2020, Yang et al., 2018] and search engines [Joachims et al., 2018]. Sometimes, one may also understand the process behind missing labels and be able to estimate propensities directly, e.g., in online advertising, true labels are received after a long delay, allowing one to train the propensity estimator from historical data [Ktena et al., 2019, Yasui et al., 2020].

However, the above-mentioned methods are usually unavailable. Therefore, it would be useful if propensities could be estimated directly from a biased training set. Unfortunately, this is an ill-defined problem because the absence of a label can be explained by either a small conditional probability of the label, a low propensity, or a combination of both. The additional assumptions needed for the propensity to be identifiable have been studied before in the areas of learning from positive and unlabeled data [Elkan and Noto, 2008], and novelty detection [Blanchard et al., 2010]. The comprehensive overview of the possible assumptions is given by Bekker and Davis [2020], where the weakest of the assumptions requires that the true distribution of negative samples for a given label cannot contain the positive distribution [Blanchard et al., 2010]. In these areas and under compatible assumptions, many methods for estimating the error ratio or label priors, both directly related to propensity estimates, were proposed [Bekker and Davis, 2020]. These include methods for estimating the unbiased conditional label probabilities and propensities jointly on the biased training set [Zhu et al., 2020, Teisseyre et al., 2020]. However, these methods often prove impractical for extreme multi-label classification (XMLC) scenarios, where most labels have very few training examples. As such, the details of these methods fall outside the scope of this work.

4.2 Empirical propensity model of Jain et al. [2016] and its relation with long tail

In this section, we will focus on a particular empirical propensity model introduced by Jain et al. [2016], which became a standard used in the evaluation of XMLC algorithms. Jain et al. [2016] proposed to model the propensities of labels as a function of their frequencies, with each label having a constant propensity value. This approach reflects an observation that less common labels are more likely to be missing, as annotators are more likely to not know or forget about uncommon labels. For example, users on a social network may not tag content with relevant but unfamiliar tags, and shoppers may overlook less popular products even when they match their needs.

Let ϕ denote a propensity model. The model defined in [Jain et al., 2016],

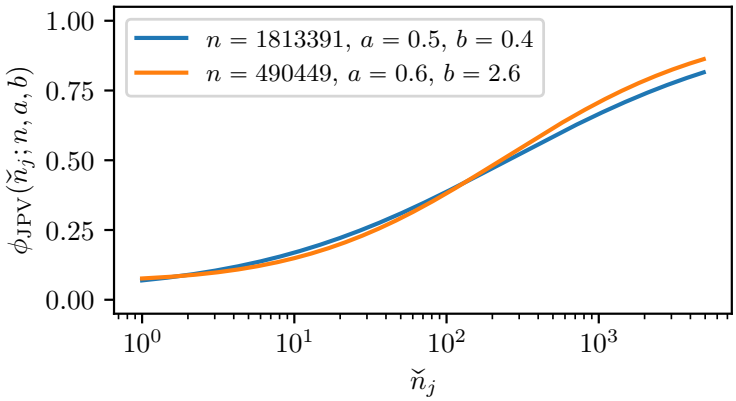


Figure 4.1: Plots of propensities calculated according to ϕ_{JPV} using parameters for Wikipedia-500K and Amazon-670K datasets.

which we denote as ϕ_{JPV},[1], for each label j can be expressed via $\breve{n}_j$ the number of training samples that are observed with label j being positive:

$$p_j = \phi_{\mathrm{JPV}}(\breve{n}_j; n, a, b) := \frac{1}{1 + (\log n - 1)(b+1)^a e^{-a \log(\breve{n}_j + b)}}, \tag{4.16}$$

where n is the total number of training samples, and a and b are dataset-dependent parameters. We demonstrate the behavior of this propensity model for two datasets and sets of parameters on Figure 4.1.

In order to arrive at this model and determine values for a and b, Jain et al. [2016] investigated two data sources that were used to derive some of the benchmark datasets in the popular XMLC Repository [Bhatia et al., 2016]. They used auxiliary information available in these data sources to identify at least some missing labels.

The first one was a collection of Wikipedia articles and their categories, that was used to create a few benchmark datasets. In these datasets, the task is to assign a set of categories based on article content. The Wikipedia categories are organized into a hierarchy that was used to estimate a and b parameters of the propensity model. Jain et al. [2016] assumed that if a label (category) is relevant to an article, then all its ancestors in the hierarchy should also be relevant. If not present, they are counted as missing. This allows for the calculation of the fraction of instances in which the label is missing over the number of instances in which it appears. We recreated this estimation procedure and obtained a plot that we present on Figure 4.2. This indeed seems to follow a sigmoidal shape produced by the ϕ_{JPV} model. The parameters a and b were then determined by fitting the model against the estimated values. Only labels with more than 4 descendants were used to improve robustness. The values obtained by Jain et al. [2016] this way are $a = 0.5, b = 0.4$.

The second source comprised Amazon's traffic, product, and sales data, which were used to create a few benchmark datasets, including item-to-item recommendation tasks. In this case, missing labels were approximated using "also viewed" and "also bought" information. Following the approach of McAuley et al. [2015],

[1] From the first letters of the authors' surnames

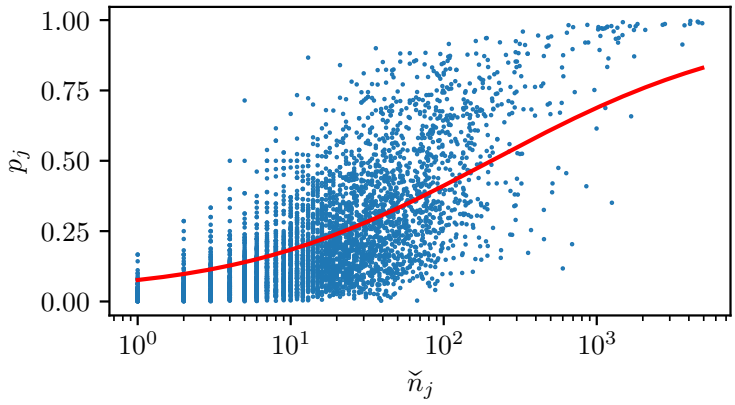


Figure 4.2: Reproduced estimates of propensities for Wikipedia-500K dataset using labels hierarchy and propensity function ϕ_{JPV} with $a = 0.5$ and $b = 0.4$ as estimated by Jain et al. [2016] for this dataset.

it was assumed that a label j (an item) is relevant to all the items viewed along with items that were also bought with the label j. Through this analysis, Jain et al. [2016] obtained values $a = 0.6$ and $b = 2.6$. Unfortunately, we find that the description provided in [Jain et al., 2016] lacks sufficient detail to allow a reliable replication of their estimation process.

For other datasets in XMLC Repository, if there is no other possibility of estimating parameters a and b, the authors proposed to use averages of the values obtained for Wikipedia and Amazon datasets (which are $a = 0.55$ and $b = 1.5$).

The propensity model of Jain et al. [2016], particularly when combined with propensity-scored precision as an evaluation metric, has become a standard in the field. This widespread adoption has persisted across numerous publications, often without questioning its rationality.

4.2.1 Shortcomings of propensity model of Jain et al. [2016]

Following the analysis presented in [Schultheis et al., 2022], in this section, we examine the validity of the Jain et al. [2016] propensity model.

Missing-labels assumption

In order to derive unbiased utilities or losses, one needs to impose precise assumptions on the process of how labels go missing, as initially discussed in Section 2.3.3. The model of Jain et al. [2016] calculates the propensities of labels based on their frequencies, resulting in a single constant propensity value per label. However, $\mathbb{P}[\breve{y}_j = 1 \,|\, y_j = 1] = p_j$ does not imply $\mathbb{P}[\breve{y}_j = 1 \,|\, y_j = 1, \boldsymbol{x}] = p_j$. To prove the unbiasedness of the general weighted instance-wise metric under ϕ_{JPV} propensity model, one needs to assume that propensities are independent of instance features. Moreover, for more complex functions, such as recall@k, this assumption may take the form of $\mathbb{P}[\breve{y}_j = 1 \,|\, y_j = 1, \boldsymbol{y}_{\neg j}, \boldsymbol{x}] = p_j$ (additional independence of other

labels), where $\boldsymbol{y}_{\neg j}$ represents ground-truth labels without label j (as it was shown in [Schultheis and Babbar, 2021]).

In general, we cannot expect the independence of missing labels from the instance features to hold. Consider, for example, cases where the feature and label space are of a similar origin [Dahiya et al., 2021], such as matching Wikipedia articles to categories. It seems unlikely that a label such as "Poland" would be missing for articles containing the word "Poland" in the subject, but it might be overlooked in articles related to Poland that do not explicitly mention the country's name.

Similarly, the assumption that the propensities do not depend on other labels going missing does not need to be held in practice, either. For example, a user who tagged the article for "Poland" with "Member states of the European Union" might be primed to think of more examples of organizations in which Poland is a member, and thus, e.g., "Current member states of the United Nations" might be less likely to be forgotten than if the EU membership had already been overlooked. However, as we already demonstrated in Section 4.1, the unbiased estimate may not actually require this dependence, if the loss function can be written as a sum over contributions from each label individually.

The assumption of ϕ_{JPV} model that propensities are constant for each label simplifies the model significantly, leading to much simpler computational procedures. Unfortunately, if this assumption is not satisfied, then one may get implausible results.

Scaling behavior

Let us observe that (4.16) does not preserve propensity estimates if the number of samples is changed without changing their characteristics, e.g., by sub- or over-sampling the dataset. In particular, if one increases the amount of available data by making multiple copies of the dataset, which should not change the estimates of label priors $\breve{\pi}_j$ given by $\breve{n}_j/n$ the JPV model will estimate propensities to be equal 1, i.e., no missing labels, as the amount of data goes to infinity:

$$\begin{aligned}\lim_{n\to\infty} \phi_{\text{JPV}}(\breve{n}_j; n, a, b) &= \frac{1}{1 + (b+1)^a \lim_{n\to\infty}(\log n - 1)e^{-a\log(\breve{n}_j + b)}} \\ &= \frac{1}{1 + (b+1)^a \lim_{n\to\infty}(\log n)\breve{n}_j^{-a}} = 1. \end{aligned} \tag{4.17}$$

This means that we cannot interpret a and b as parameters of some underlying (unknown) process that describes the labeling process, as we cannot even fix a and b when the data comes from the same process.

Parameters estimation process and dependence on frequencies

Setting aside structural concerns about (4.16), the choice of a model and estimation of the parameters a and b still remains an issue. The authors' approach of using additional information to identify missing labels can only establish an upper bound on propensities, as labels might be missing through other mechanisms. For

Table 4.1: Normalized and unnormalized propensity-scored precision of PfastreXML [Jain et al., 2016], when using the JPV model, with $a = 0.5$, $b = 0.4$ for WikiLSHTC-325K and $a = 0.6$, $b = 2.6$ for Amazon-670K.

	WikiLSHTC-325K			Amazon-670K		
PSP(%)	@1	@3	@5	@1	@3	@5
Normalized	31.16	31.80	33.35	29.93	31.26	32.80
Unnormalized	196.96	118.54	85.28	326.47	282.28	250.57

example, in the case of Wikipedia data, only about 40000 out of 500000 labels met the criteria for having sufficient descendants, while approximately 300000 labels had no descendants at all, making it impossible to evaluate them under this protocol.

Even if we ignore labels that cannot be used and assume that the upper bound is roughly close to the true propensity, the choice of the model remains a question of its own. Although there is clearly a trend that labels within a given frequency range have, on average, a certain propensity, for each individual label, the actual propensity can fluctuate widely around this mean, as shown on Figure 4.2.

Finally, the selection of a and b parameters remains problematic, Jain et al. [2016] did not describe details of fitting parameters a and b to estimate per-label propensities, as such, we do not know the objective function of the fitting process, and whether the fitting was performed on propensities or their inverse $\frac{1}{p_j}$ - an important distinction given their typical appearance in unbiased utility and loss formulations.

Implausible results and normalization

While we cannot directly verify the assumptions and validity of the JPV model without clean ground-truth data, we can demonstrate that the approach of Jain et al. [2016] produces theoretically inconsistent results. For example, propensity-scored precision@k, as an unbiased estimate of precision@k on the ground-truth data, should be bounded between zero and one. However, empirical calculations for actual classifiers often yield values substantially outside this range. Of course, for an individual instance or a small subset of them, the unbiased estimate does not need to fall into that range, but a large deviation from the true value becomes exceedingly unlikely when averaging over the entire dataset.

To circumvent this issue, Jain et al. [2016] suggest to report a normalized version of propensity-scored precision@k, also calling this measure "propensity-scored precision". The normalization is realized by dividing the metrics' value by the largest possible value that any prediction could have achieved on that data:

$$\text{Normalized PSP@}k = \frac{\sum_{i=1}^{n} \text{PSP@}k(\boldsymbol{y}_i, \widehat{\boldsymbol{y}}_i)}{\sum_{i=1}^{n} \max_{\boldsymbol{z}} \text{PSP@}k(\boldsymbol{y}_i, \boldsymbol{z})} \,. \tag{4.18}$$

Table Table 4.1 reports the values of both variants of propensity-scored precision@k, showing how severe this issue is.

The normalization introduces a factor that is constant over the entire dataset,

and thus does not influence model selection. However, it removes the interpretation of the received value as an unbiased estimate of the metric on clean data, and it hides the model misspecification. Unfortunately, in subsequent literature, the distinction between the unbiased metrics and the normalized versions are not always preserved, e.g., Bhatia et al. [2016] present unbiased formulas but lists normalized values.

4.2.2 Relation to long tails

The form of (4.16) assigns lower propensities to tail labels. Consequently, under an unbiased metric using ϕ_{JPV}, correct predictions of tail labels receive higher weights than head labels. In particular, the resulting weightings resemble other weightings schemes used for long-tailed or unbalanced learning tasks, leading the authors to conclude:

> "Such weights arise naturally as inverse propensities in the unbiased losses developed in this paper. [...] This not only provides a sound theoretical justification of label weighting heuristics for recommending rare items but also leads to a more principled setting of the weights."

As a result, propensity-scored variants ended up being viewed as metrics in their own right and are currently used for both counteracting missing labels (as unbiased estimates) and to weigh tail labels (as independent metrics), becoming established performance metrics commonly used in XMLC. We list several examples of references to propensity-scored losses:

- "We examined the performance on tail labels by PSP@k" [You et al., 2019a];
- "We achieve high precision and propensity scores, thus demonstrating the effectiveness of our method even on infrequent tail labels." [Guo et al., 2019];
- "capture prediction accuracy of a learning algorithm at top-k slots of prediction, and also the diversity of prediction by giving a higher score for predicting rarely occurring tail-labels" [Babbar and Schölkopf, 2019];
- "propensity scored precision@k, which has recently been shown to be an unbiased, and more suitable, metric" [Jain et al., 2019];
- "which leads to better performance on tail labels." [Yen et al., 2017];
- "propensity scored variant, which is unbiased and assigns higher rewards for accurate tail label predictions",
- "evaluate prediction performance on tail labels using propensity scored variants" [Khandagale et al., 2020];
- "replacing the nDCG loss with its propensity scored variant and using additional classifiers designed for tail labels" [Tagami, 2017a].

We argue that this usage conflates two distinct issues: correcting metrics under missing labels and promoting tail labels. In the missing-labels context, propensity adjustment represents the correct method for calculating the true performance of

a classifier, not an alternative metric. Only in the tail-performance interpretation, it does make sense to speak of a trade-off in performance between vanilla and propensity-scored metrics, with one focusing on the head and another on tail labels. However, then the weights lose their original interpretation and are reduced to arbitrary tail-label adjustment, making it similar to other simple propensity models that have been introduced in other domains [Saito et al., 2020, Yang et al., 2018, Joachims et al., 2018]. For example, a propensity model used frequently in recommendation systems is given by the following power-law formulation:

$$p_j = \phi(\breve{\pi}_j; \alpha, \beta) := (\alpha \breve{\pi}_j)^{-\beta} \ , \tag{4.19}$$

where $\breve{\pi}_j := \mathbb{P}[\breve{y} = 1]$ is observed prior probability of label j and α being often set to $\max_j \breve{n}_j/n$ [Yang et al., 2018, Saito et al., 2020]. This model, similar to JPV, assigns higher weights to less frequent labels.

For the use case of measuring tail-label performance, it would be preferable to have a metric that treats tail labels in a principled way. Of course, in XMLC, both interpretations can be combined, i.e., one would like to have a task utility that is adapted to tail labels, but calculate it in a way that takes missing labels into account.

4.3 Summary of the chapter

In this chapter, we've first derived unbiased variants of general instance-wise metrics at k under missing label settings. However, the usage of these unbiased estimates requires knowledge of propensities, and accurate estimation of these values remains a challenging problem. Next, we've discussed the propensity model and accompanying propensity-scored metrics introduced by Jain et al. [2016] that have emerged as a standard approach for evaluating XMLC algorithms. Our closer examination revealed several theoretical and practical limitations of this model. Moreover, the current usage of propensity-scored metrics in the field reveals a concerning conflation of two distinct challenges: missing labels and tail labels.

The work of Jain et al. [2016] represented an important step forward in XMLC evaluation. The model's widespread adoption, despite its issues and limitations, indicates a need for better solutions to handle both missing and long-tail labels in XMLC tasks. Looking forward, we believe the field would benefit from a clearer separation between methods for handling missing labels and approaches for addressing tail labels. The feature work could focus on developing more robust propensity models and estimation methods. However, we believe the development of dedicated metrics for tail label performance that do not rely on propensity scoring could help disentangle these distinct challenges in XMLC evaluation. Because of that, in the following chapters, we explore a family of metrics that may provide an alternative for evaluating tail-label performance.

5
Label-wise metrics at k

Following the instance-wise measures, in this chapter, we introduce a more general family of complex metrics defined on the confusion matrix, which are linearly decomposable over labels, and as such, we call them label-wise utilities. This family includes macro-averaged utilities, but also the general instance-wise weighted utilities discussed in the previous chapter.

5.1 Metrics linearly decomposable over labels

We start with a definition of the general class of utilities, linearly decomposable over labels, that are expressed using a multi-label confusion matrix:

$$\begin{aligned}\Psi\left(\widehat{\mathbf{C}}\left(\mathbf{Y}, \widehat{\mathbf{Y}}^{@k}\right)\right) &= \sum_{j=1}^{m} \psi^j\left(\widehat{\boldsymbol{c}}\left(\boldsymbol{y}_{:,j}, \widehat{\boldsymbol{y}}_{:,j}\right)\right) \\ &= \sum_{j=1}^{m} \psi^j\left(\widehat{c}_{j,\mathrm{tn}}, \widehat{c}_{j,\mathrm{fp}}, \widehat{c}_{j,\mathrm{fn}}, \widehat{c}_{j,\mathrm{tp}}\right) , \end{aligned} \tag{5.1}$$

where ψ^j is a binary utility function that is defined on the entries of the binary confusion matrix for a single label j, and might be nonlinear in itself. We allow this function to be different for each label j. Below, we demonstrate how this class of utilities covers a subset of instance-wise weighted utilities and macro-averaged utilities.

5.2 Instance-wise weighted utility functions as metrics linearly decomposable over labels

The instance-wise weighted utility functions that we introduced in (3.9) can be seen as a special case of (5.1). To calculate the corresponding task utility for a given $\mathbf{Y}$ and $\widehat{\mathbf{Y}}$, the instance-wise utility is averaged over all samples. By interchanging the order of summation, we can see that it is of the form (5.1):

$$\begin{aligned}\Psi_{\text{instance}}\left(\mathbf{Y}, \widehat{\mathbf{Y}}^{@k}\right) &= \frac{1}{n}\sum_{i=1}^{n} \mathrm{U}\left(\boldsymbol{y}_j, \widehat{\boldsymbol{y}}_j^{@k}\right) \\ &= \sum_{i=j}^{m} \boldsymbol{g}_j \cdot \widehat{\boldsymbol{c}}(\boldsymbol{y}_{:,j}, \widehat{\boldsymbol{y}}_{:,j}) \\ &= \sum_{i=j}^{m} g_{j,\text{tn}}\widehat{c}_{j,\text{tn}} + g_{j,\text{fp}}\widehat{c}_{j,\text{fp}} + g_{j,\text{fn}}\widehat{c}_{j,\text{fn}} + g_{j,\text{tp}}\widehat{c}_{j,\text{tp}} \\ &= \sum_{j=1}^{m} \psi^j\left(\widehat{\boldsymbol{c}}\left(\boldsymbol{y}_{:,j}, \widehat{\boldsymbol{y}}_{:,j}^{@k}\right)\right) . \end{aligned} \tag{5.2}$$

Thus, we see that instance-wise weighted utilities can be seen as linear confusion matrix-based metrics, because of this, we will refer to them as linear utilities in this chapter.

Since $\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{g}_j \cdot \widehat{\boldsymbol{c}}(\boldsymbol{y}_{:,j}, \widehat{\boldsymbol{y}}_{:,j}) = \frac{1}{m}\mathbf{G} \cdot \widehat{\mathbf{C}}(\boldsymbol{y}, \widehat{\boldsymbol{y}})$, the instance-wise weighted utilities are clearly linear w.r.t. the confusion matrix and the form of the optimal classifier for utilities of this type is the same in EIU, ETU, and PU frameworks and it is of the form presented in Theorem 3.2.1.

5.3 Macro-average of non-decomposable utilities

In the following chapters, we are more interested in optimizing performance metrics that do not decompose into individual instances, but are general functions of the confusion matrix of classifier $\boldsymbol{h}$ that are linearly decomposable over labels as macro-averaged multi-label metrics. The macro-averaged utilities report the value of a binary classification utility ψ, averaged over all labels. This corresponds to setting $\psi^j = \frac{1}{m}\psi$ for all labels j in (5.1), yielding:

$$\Psi_{\text{macro}}\left(\widehat{\mathbf{C}}\left(\mathbf{Y}, \widehat{\mathbf{Y}}^{@k}\right)\right) = \frac{1}{m}\sum_{j=1}^{m} \psi\left(\widehat{\boldsymbol{c}}\left(\boldsymbol{y}_{:,j}, \widehat{\boldsymbol{y}}_{:,j}^{@k}\right)\right) . \tag{5.3}$$

This, in turn, can be used to, for example, define macro-precision@k, macro-recall@k, macro-F_β-measure@k, and coverage@k[1], that we already mentioned in Chapter 1.

[1]To shorten utility names, we often refer to macro-averaged variants of specific metrics by adding just the "macro-" prefix.

Table 5.1: Examples of macro-averaged performance metrics, with their form as $\psi(\mathrm{tn}, \mathrm{fp}, \mathrm{fn}, \mathrm{tp})$ and $\psi(\mathrm{tp}, \mathrm{pp}, \mathrm{cp})$.

Metric	$\psi(\mathrm{tn}, \mathrm{fp}, \mathrm{fn}, \mathrm{tp})$	$\psi(\mathrm{tp}, \mathrm{pp}, \mathrm{cp})$
Hamming score	$\mathrm{tp} + \mathrm{tn}$	$1 + 2\mathrm{tp} - \mathrm{pp} - \mathrm{cp}$
Balanced accuracy	$\frac{\mathrm{tp}}{2(\mathrm{tp}+\mathrm{fn})} + \frac{\mathrm{tn}}{2(\mathrm{tn}+\mathrm{fp})}$	$\frac{\mathrm{tp}}{2\mathrm{cp}} + \frac{1+\mathrm{tp}-\mathrm{pp}-\mathrm{cp}}{2(1-\mathrm{cp})}$
Precision	$\frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fp}}$	$\frac{\mathrm{tp}}{\mathrm{pp}}$
Recall	$\frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fn}}$	$\frac{\mathrm{tp}}{\mathrm{cp}}$
F_β-measure	$\frac{(1+\beta^2)\mathrm{tp}}{(1+\beta^2)\mathrm{tp}+\beta^2\mathrm{fn}+\mathrm{fp}}$	$\frac{(1+\beta^2)\mathrm{tp}}{\beta^2\mathrm{cp}+\mathrm{pp}}$
Jaccard similarity	$\frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fp}+\mathrm{fn}}$	$\frac{\mathrm{tp}}{\mathrm{cp}+\mathrm{pp}-\mathrm{tp}}$
Coverage	$\mathbb{1}\left[\mathrm{tp} > 0\right]$	$\mathbb{1}\left[\mathrm{tp} > 0\right]$

We show the form of ψ for these and other popular measures in Table 5.1 for both presented parameterizations of the confusion matrix in Section 2.2.1 (using true negatives, false negatives, false positives, and true positives, as well as the alternative using true positives, predicted positives, and positive label ratios). Because of the averaging of ψ over labels, the resulting utilities emphasize the balance between labels, independently of their frequencies, and potentially alleviate the problems with evaluating "long-tail" performance. On Figure 5.1 we present the comparison of instance-wise and macro-averaged precision@k for a small example.

Micro-averaged metrics defined on the multi-label confusion matrix

Using the same notation of the multi-label confusion matrix (2.5), we can also define another class of metrics called micro-averaged:

$$\Psi_{\mathrm{micro}}(\widehat{\mathbf{C}}(\mathbf{Y}, \widehat{\mathbf{Y}})) = \psi\left(\frac{1}{m}\sum_{j=1}^{m}\widehat{c}_{j,\mathrm{tn}}, \frac{1}{m}\sum_{j=1}^{m}\widehat{c}_{j,\mathrm{fp}}, \frac{1}{m}\sum_{j=1}^{m}\widehat{c}_{j,\mathrm{fn}}, \frac{1}{m}\sum_{j=1}^{m}\widehat{c}_{j,\mathrm{tp}}\right). \tag{5.4}$$

Since the micro-averaged utilities do not balance the contribution of labels to the final value, they are rarely used in XMLC and are of less interest in the context of this thesis. We very briefly discuss the optimization of micro-metrics without a budget constraint at the end of this chapter. When it comes to micro-averaged metrics with a budget @k, the methods presented in this thesis can also be applied for their optimization.

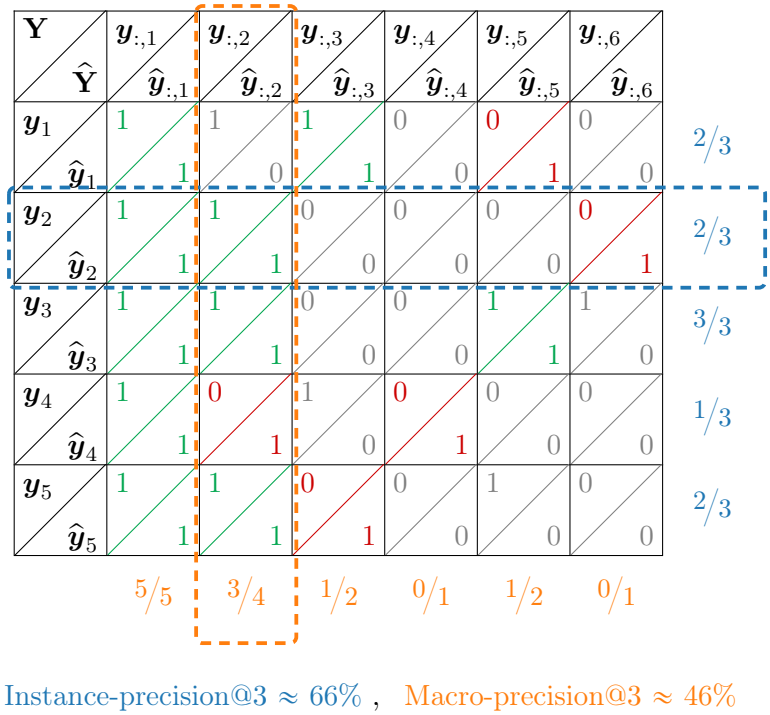


Figure 5.1: An example of calculating instance- and macro-precision@3 for given true label matrix $\mathbf{Y}$ and predicted labels $\widehat{\mathbf{Y}}$ with $n = 5$ and $m = 6$.

5.4 Difficulty of optimization under budget at k

Even though the objective (5.1) decomposes into m binary problems, these are still coupled by the budget constraint, $\|\boldsymbol{h}(\boldsymbol{x})\|_1 = k$ for all $\boldsymbol{x} \in \mathcal{X}$, and cannot be optimized independently. To demonstrate this coupling effect, we present a simple illustrative example. We consider two performance measures: the Hamming score and the macro-Jaccard similarity, as defined in Table 5.1 without or with a budget $k = 2$. Additionally, we introduce two simple distributions, b, each consisting of two distinct instances, each with an equal probability of 50% and three possible labels. We can treat this example as a problem of optimization for the population, as in the PU setting, or for a given test set with these two instances, as in the ETU setting. In both interpretations, the results and conclusions remain the same.

The example is presented in Figure 5.2. Notice that the only difference between the two distributions is the marginal conditional probability of the third label for the second instance $\boldsymbol{x}_2$ (i.e., $\eta_3(\boldsymbol{x}_2)$). First, we examine the Hamming score and macro-Jaccard similarity without a budget k. The lack of budget constraints makes the problem for each label and instance independent, and indeed, the optimal prediction for both metrics differs only for the label, with different probabilities between distributions A and B. Next, we consider the Hamming score with a budget $k = 2$. This time, the difference between optimal prediction varies not only

on the label with different probabilities, but the budget constraint forces us to also change the other label for the same instance. However, because of the linearity of the Hamming score, prediction for the other instances is not affected. Finally, for macro-Jaccard similarity with budget k, we observe that despite the difference in one marginal conditional probability, the optimal solution is also different on the other instance for the two other labels, which highlights the coupling effect of budget k combined with nonlinear matrices. If it were possible to find a solution for each label separately, the change in the distribution on one label would not affect the other labels in other instances.

5.5 Summary of the chapter

In this chapter, we have introduced a general class of label-wise metrics that are linearly decomposable over labels. We show that this class covers instance-wise weighted utilities introduced in the previous Chapter 3 (e.g., precision@k, Hamming score@k and their weighted variants) and macro-averaged utilities that are attractive in the context of long tail distributions of labels, as they emphasize the equal importance between labels, independently of their frequencies. We also demonstrated that this class of utilities, under the constraint of predictions budgeted at k, cannot be optimized by decomposing the problem into separate binary problems, as in the unconstrained cases, which have already been well studied in the literature. In Chapters 6 and 7, we will focus on the optimization of this general class of metrics in the ETU and PU frameworks, respectively, under the budget at k.

Optimization of measures without a budget at k

When optimizing the macro-average without a budget at k constraint, the problem decomposes into independent problems of optimizing the binary utility for each label separately [Koyejo et al., 2015, Kotłowski and Dembczyński, 2017]. This problem is well studied in the literature. Interestingly, in both frameworks, the optimal solution is based on thresholding marginal conditional label probabilities [Dembczyński et al., 2017], but the resulting thresholds are different, with the discrepancy diminishing with the size of the test set. The threshold tuning for PU is usually performed on a validation set [Yang, 2001, Lin and Lin, 2023], while the exact optimization for ETU is performed on a test set. It requires cubic time in the general case and quadratic time in some special cases [Ye et al., 2012, Natarajan et al., 2016a]. Approximate solutions can be obtained in linear time [Lewis, 1995, Dembczyński et al., 2017]. The situation is similar in the case of micro-averaged metrics, with the key difference being that the optimal threshold is determined by a single value, the same for all the labels [Koyejo et al., 2015, Kotłowski and Dembczyński, 2017].

Distribution A:

	$\mathbb{P}[\boldsymbol{x}]$	$\eta_1(\boldsymbol{x})$	$\eta_2(\boldsymbol{x})$	$\eta_3(\boldsymbol{x})$
$\boldsymbol{x}_1$	0.5	0.4	0.2	0.6
$\boldsymbol{x}_2$	0.5	0.8	0.4	**0.3**

Distribution B:

	$\mathbb{P}[\boldsymbol{x}]$	$\eta_1(\boldsymbol{x})$	$\eta_2(\boldsymbol{x})$	$\eta_3(\boldsymbol{x})$
$\boldsymbol{x}_1$	0.5	0.4	0.2	0.6
$\boldsymbol{x}_2$	0.5	0.8	0.4	**0.8**

Optimization of macro Hamming score without budget k:

Optimal $\boldsymbol{h}_A^\star$ for distribution A:

	$\widehat{y}_1$	$\widehat{y}_2$	$\widehat{y}_3$
$\boldsymbol{x}_1$	0	0	1
$\boldsymbol{x}_2$	1	0	**0**

$\Psi_{\text{Hamming}}\left(\widehat{\mathbf{C}}(\boldsymbol{h}_A^\star, A)\right) \approx 68.333\%$

Optimal $\boldsymbol{h}_B^\star$ for distribution B:

	$\widehat{y}_1$	$\widehat{y}_2$	$\widehat{y}_3$
$\boldsymbol{x}_1$	0	0	1
$\boldsymbol{x}_2$	1	0	**1**

$\Psi_{\text{Hamming}}\left(\widehat{\mathbf{C}}(\boldsymbol{h}_B^\star, B)\right) \approx 70.000\%$

Optimization of macro Jaccard similarity without budget k:

Optimal $\boldsymbol{h}_A^\star$ for distribution A:

	$\widehat{y}_1$	$\widehat{y}_2$	$\widehat{y}_3$
$\boldsymbol{x}_1$	1	0	1
$\boldsymbol{x}_2$	1	1	**0**

$\Psi_{\text{Macro-Jaccard}}\left(\widehat{\mathbf{C}}(\boldsymbol{h}_A^\star, A)\right) \approx 46.496\%$

Optimal $\boldsymbol{h}_B^\star$ for distribution B:

	$\widehat{y}_1$	$\widehat{y}_2$	$\widehat{y}_3$
$\boldsymbol{x}_1$	1	0	1
$\boldsymbol{x}_2$	1	1	**1**

$\Psi_{\text{Macro-Jaccard}}\left(\widehat{\mathbf{C}}(\boldsymbol{h}_B^\star, B)\right) \approx 54.444\%$

Optimization of Hamming score with budget $k = 2$:

Optimal $\boldsymbol{h}_A^\star$ for distribution A:

	$\widehat{y}_1$	$\widehat{y}_2$	$\widehat{y}_3$
$\boldsymbol{x}_1$	0	1	1
$\boldsymbol{x}_2$	1	**1**	**0**

$\Psi_{\text{Hamming@2}}\left(\widehat{\mathbf{C}}(\boldsymbol{h}_A^\star, A)\right) \approx 61.667\%$

Optimal $\boldsymbol{h}_B^\star$ for distribution B:

	$\widehat{y}_1$	$\widehat{y}_2$	$\widehat{y}_3$
$\boldsymbol{x}_1$	0	1	1
$\boldsymbol{x}_2$	1	**0**	**1**

$\Psi_{\text{Hamming@2}}\left(\widehat{\mathbf{C}}(\boldsymbol{h}_B^\star, B)\right) \approx 66.667\%$

Optimization of macro Jaccard similarity with budget $k = 2$:

Optimal $\boldsymbol{h}_A^\star$ for distribution A:

	$\widehat{y}_1$	$\widehat{y}_2$	$\widehat{y}_3$
$\boldsymbol{x}_1$	**1**	**0**	1
$\boldsymbol{x}_2$	1	**1**	**0**

$\Psi_{\text{Macro-Jaccard@2}}\left(\widehat{\mathbf{C}}(\boldsymbol{h}_A^\star, A)\right) \approx 46.496\%$

Optimal $\boldsymbol{h}_B^\star$ for distribution B:

	$\widehat{y}_1$	$\widehat{y}_2$	$\widehat{y}_3$
$\boldsymbol{x}_1$	**0**	**1**	1
$\boldsymbol{x}_2$	1	**0**	**1**

$\Psi_{\text{Macro-Jaccard@2}}\left(\widehat{\mathbf{C}}(\boldsymbol{h}_B^\star, B)\right) \approx 47.143\%$

Figure 5.2: An example demonstrating the coupling effect of predictions budgeted at k for utilities non-decomposable over instances.

6

Optimization of label-wise metrics at k under expected test utility framework

In this chapter, we analyze the problem of optimizing label-wise metrics at k under the expected test utility (ETU) framework, and propose an efficient algorithm for this task. This chapter mostly summarizes the results published in Schultheis et al. [2023].

6.1 Order-invariant utilities are confusion matrix utilities

Let us first recall that in the ETU framework, we are interested in the expected utility of a classifier on a provided test set (2.9) (see Section 2.2.3 for the definition of the problem):

$$\Phi_{\text{ETU}}(\boldsymbol{h}) := \mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \mid \mathbf{X}]} \left[\Psi \left(\widehat{\mathbf{C}} \left(\mathbf{Y}, \boldsymbol{h}(\mathbf{X}) \right) \right) \right] = \mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \mid \mathbf{X}]} \left[\Psi \left(\widehat{\mathbf{C}} \left(\mathbf{Y}, \widehat{\mathbf{Y}} \right) \right) \right], \tag{6.1}$$

Notice that ETU does not strictly require a utility to be defined on the confusion matrix. Because of that, in this section, we often just write $\Psi(\mathbf{Y}, \widehat{\mathbf{Y}})$ instead of $\Psi \left(\widehat{\mathbf{C}} \left(\mathbf{Y}, \widehat{\mathbf{Y}} \right) \right)$. It is worth noting here that any utility function of the form:

$$\Psi(\mathbf{Y}, \widehat{\mathbf{Y}}) = \sum_{j=1}^{m} \psi^j(\boldsymbol{y}_{:,j}, \widehat{\boldsymbol{y}}_{:,j}), \tag{6.2}$$

which is invariant under instance reordering (i.e., its value does not change if rows of both matrices are re-ordered using the same permutation), can be defined in terms of the multi-label confusion matrix (2.5). We present a formal proof in Appendix A.3.1 for a more general form:

$$\Psi \left(\mathbf{Y}, \widehat{\mathbf{Y}} \right) = f \left(\psi^1(\boldsymbol{y}_{:,1}, \widehat{\boldsymbol{y}}_{:,1}), \psi^2(\boldsymbol{y}_{:,2}, \widehat{\boldsymbol{y}}_{:,2}), \ldots, \psi^m(\boldsymbol{y}_{:,m}, \widehat{\boldsymbol{y}}_{:,m}) \right), \tag{6.3}$$

where f is any aggregation function, not only a sum as in (6.2).

6.2 Sufficiency of label probability estimates

Let us observe that label probabilities $\boldsymbol{\eta}$ are sufficient to make optimal predictions according to (2.10). With the assumption that the labels of the specific instance are independent of other instances, $\mathbb{P}[\mathbf{Y}|\mathbf{X}] = \prod_{i=1}^{n} \mathbb{P}[\boldsymbol{y}_i|\boldsymbol{x}_i]$, we obtain:

$$\begin{aligned}
\Phi_{\mathrm{ETU}}(\boldsymbol{h}) &= \mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\Psi(\mathbf{Y}, \boldsymbol{h}(\mathbf{X}))\right] = \mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\Psi\left(\mathbf{Y}, \widehat{\mathbf{Y}}\right)\right] \\
&= \sum_{j=1}^{m} \mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\psi^j(\boldsymbol{y}_{:,j}, \widehat{\boldsymbol{y}}_{:,j})\right] \\
&= \sum_{j=1}^{m} \sum_{\boldsymbol{y}'\in\{0,1\}^n} \mathbb{P}[\boldsymbol{y}_{:,j} = \boldsymbol{y}' \,|\, \mathbf{X}]\psi^j(\boldsymbol{y}', \widehat{\boldsymbol{y}}_{:,j}) \\
&= \sum_{j=1}^{m} \sum_{\boldsymbol{y}'\in\{0,1\}^n} \left(\prod_{i=1}^{n} \mathbb{P}[y_{i,j} = y'_i \,|\, \boldsymbol{x}_i]\right) \psi^j(\boldsymbol{y}', \widehat{\boldsymbol{y}}_{:,j}) \\
&= \sum_{j=1}^{m} \sum_{\boldsymbol{y}'\in\{0,1\}^n} \left(\prod_{i=1}^{n} \eta_j(\boldsymbol{x}_i)y'_i + (1-\eta_j(\boldsymbol{x}_i))(1-y'_i)\right) \psi^j(\boldsymbol{y}', \widehat{\boldsymbol{y}}_{:,j}) \qquad (6.4)
\end{aligned}$$

This equation lays out a daunting optimization task, which requires summing over 2^n many summands $\boldsymbol{y}'$. In the case of binary classification, there exist methods to solve the problem exactly in $\mathcal{O}(n^3)$, or in $\mathcal{O}(n^2)$ in some special cases [Natarajan et al., 2016a]. By using semi-empirical quantities (defined below), one can design approximate algorithms that run in $\mathcal{O}(n)$ [Lewis, 1995, Dembczyński et al., 2017]. Following this approach, we introduce a semi-empirical ETU approximation for label-wise metrics.

If this approximation results in a linear function of test instances, the problem decomposes and can be solved easily. Otherwise, we propose to use an algorithm that leads to locally optimal predictions.

6.3 Semi-empirical ETU approximation

For the following sections in this chapter, to make notation more compact, we switch to the alternative parametrization of the confusion matrix by the ratios of true positives (tp), predicted positives (pp) and conditional positives (cp), as introduced in (2.7).

As we showed in (6.4), in order to compute Φ_{ETU}, one needs to take into account every possible combination of confusion-matrix values, and calculate the corresponding value of Ψ, which is then averaged according to the respective probabilities. A computationally easier approach is to average the predictions first, to get a single value for the confusion matrix, and calculate Ψ for it, leading to semi-empirical quantities of true positives, predicted positives, and conditional

positives, that we denote using a tilde symbol:

$$\begin{aligned}
\tilde{\boldsymbol{c}}_{:,\mathrm{tp}} &\coloneqq \mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \mid \mathbf{X}]}[\widehat{\boldsymbol{c}}_{:,\mathrm{tp}}] = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\eta}(\boldsymbol{x}_i) \odot \widehat{\boldsymbol{y}}_i \,, \\
\tilde{\boldsymbol{c}}_{:,\mathrm{pp}} &\coloneqq \mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \mid \mathbf{X}]}[\widehat{\boldsymbol{c}}_{:,\mathrm{pp}}] = \frac{1}{n} \sum_{i=1}^{n} \widehat{\boldsymbol{y}}_i \,, \\
\tilde{\boldsymbol{c}}_{:,\mathrm{cp}} &\coloneqq \mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \mid \mathbf{X}]}[\widehat{\boldsymbol{c}}_{:,\mathrm{cp}}] = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\eta}(\boldsymbol{x}_i) \,,
\end{aligned} \tag{6.5}$$

where $\tilde{\boldsymbol{c}}_{:,\mathrm{pp}} = \widehat{\boldsymbol{c}}_{:,\mathrm{pp}}$ follows because the number of predicted positives depends only on the predictions $\widehat{\mathbf{Y}}$, and $\tilde{\boldsymbol{c}}_{:,\mathrm{cp}}$ is a constant that does not depend on predictions.

This allows us to define the semi-empirical expected value of ETU:

$$\tilde{\Phi}_{\mathrm{ETU}}(\boldsymbol{h}) \coloneqq \Psi(\tilde{\boldsymbol{c}}_{:,\mathrm{tp}}, \widehat{\boldsymbol{c}}_{:,\mathrm{pp}}, \tilde{\boldsymbol{c}}_{:,\mathrm{cp}}) \,, \tag{6.6}$$

which tends to approximate

$$\Phi_{\mathrm{ETU}}(h) = \mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \mid \mathbf{X}]} \left[\Psi(\widehat{\boldsymbol{c}}_{:,\mathrm{tp}}, \widehat{\boldsymbol{c}}_{:,\mathrm{pp}}, \widehat{\boldsymbol{c}}_{:,\mathrm{cp}}) \right] . \tag{6.7}$$

Note that if Ψ is linear in all arguments depending on the random variable $\mathbf{Y}$, then the approximation is exact due to the linearity of expectations. Aside from instance-wise measures, which we showed to be linear in (5.2), the approximation is also exact for the more general class of functions with binary utilities of the form:

$$\psi^j(\mathrm{tp}, \mathrm{pp}, \mathrm{cp}) = f_j^{\mathrm{tp}}(\mathrm{pp}) \cdot \mathrm{tp} + f_j^{\mathrm{pp}}(\mathrm{pp}) + f_j^{\mathrm{cp}}(\mathrm{pp}) \cdot \mathrm{cp} \,. \tag{6.8}$$

An important example is macro-precision, with $f_j^{\mathrm{tp}}(\mathrm{pp}) = \frac{1}{m \cdot \mathrm{pp}}$ and $f_j^{\mathrm{pp}}(\mathrm{pp}) = f_j^{\mathrm{cp}}(\mathrm{pp}) = 0$ for all $j \in [m]$.

For general utilities Φ_{ETU}, the approximation $\tilde{\Phi}_{\mathrm{ETU}}$ may differ from the actual objective Φ_{ETU}. However, if we are able to control the change of the utility under small changes in its arguments, the approximation remains close to the true objective and decreases with the test set size n. To this end, following Dembczyński et al. [2017], we assume that each ψ^j is a cp-Lipschitz function, with respect to its arguments $(\mathrm{tp}, \mathrm{pp}, \mathrm{cp})$, where the Lipschitz constants may depend on cp.

Definition 6.3.1 (cp-Lipschitz [Dembczyński et al., 2017])**.** A binary classification metric $\psi(\mathrm{tp}, \mathrm{pp}, \mathrm{cp})$ is said to be cp-Lipschitz if

$$\begin{aligned}
|\psi(\mathrm{tp}, \mathrm{pp}, \mathrm{cp}) - \psi(\mathrm{tp}', \mathrm{pp}', \mathrm{cp}')| \leqslant L_{\mathrm{tp}}(\mathrm{cp})|\mathrm{tp} - \mathrm{tp}'| + L_{\mathrm{pp}}(\mathrm{cp})|\mathrm{pp} - \mathrm{pp}'| \\
+ L_{\mathrm{cp}}(\mathrm{cp})|\mathrm{cp} - \mathrm{cp}'|,
\end{aligned} \tag{6.9}$$

for any $\mathrm{pp}, \mathrm{pp}' \in [0,1]$, $\mathrm{cp}, \mathrm{cp}' \in (0,1)$, $0 \leqslant \mathrm{tp} \leqslant \min(\mathrm{cp}, \mathrm{pp})$, and $0 \leqslant \mathrm{tp}' \leqslant \min(\mathrm{cp}', \mathrm{pp}')$. The constants $L_{\mathrm{tp}}(\mathrm{cp}), L_{\mathrm{pp}}(\mathrm{cp}), L_{\mathrm{cp}}(\mathrm{cp})$ are allowed to depend on cp, in contrast to the standard Lipschitz functions.

In the Appendix A.3.2, we show that most of the metrics of interest satisfy the cp-Lipschitz assumption, including the linear confusion-matrix measures (5.2) with fixed weights (e.g., Hamming utility, instance-precision) and almost all macro

metrics listed in Table 5.1, like macro-recall and macro-F-measure, but with notable exceptions of macro-precision and coverage. Under this assumption, we can bound the approximation error.

Theorem 6.3.2. *Let each ψ^j be* cp*-Lipschitz with constants $L^j_{\mathrm{tp}}(\mathrm{cp})$, $L^j_{\mathrm{pp}}(\mathrm{cp})$, $L^j_{\mathrm{cp}}(\mathrm{cp})$. For any $\widehat{\mathbf{Y}}$ it holds:*

$$\left|\Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\mathbf{X}) - \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\mathbf{X})\right| \leqslant \frac{1}{2\sqrt{n}}\left(\sum_{j=1}^{m}\left(L^j_{\mathrm{tp}}(\tilde{c}_{j,\mathrm{cp}}) + L^j_{\mathrm{cp}}(\tilde{c}_{j,\mathrm{cp}})\right)\right). \quad (6.10)$$

Thus, using $\tilde{\Phi}_{\mathrm{ETU}}$ as a surrogate for Φ_{ETU} leads only to $\mathcal{O}(1/\sqrt{n})$ error, diminishing with the test size, while substantially simplifying the optimization process.

Proof (sketch, full proof in Appendix A.3.3). Using definitions of expected ETU utility (6.4) and its semi-empirical approximation (6.6) and applying Jensen's inequality, we get:

$$\begin{aligned}&\left|\Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\mathbf{X}) - \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\mathbf{X})\right| \\ &\quad\leqslant \sum_{j=1}^{m}\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\left|\psi^j(\widehat{c}_{j,\mathrm{tp}},\widehat{c}_{j,\mathrm{pp}},\widehat{c}_{j,\mathrm{cp}}) - \psi^j(\tilde{c}_{j,\mathrm{tp}},\widehat{c}_{j,\mathrm{pp}},\tilde{c}_{j,\mathrm{cp}})\right|\right], \quad (6.11)\end{aligned}$$

Each term in the sum can be now bounded by $(L^j_{\mathrm{tp}} + L^j_{\mathrm{cp}})/(2\sqrt{n})$ (for this analysis we drop (cp) in notation). For each $j \in [m]$, using cp-Lipschitzness (6.9) of ψ^j we have:

$$\left|\psi^j(\widehat{c}_{j,\mathrm{tp}},\widehat{c}_{j,\mathrm{pp}},\widehat{c}_{j,\mathrm{cp}}) - \psi^j(\tilde{c}_{j,\mathrm{tp}},\widehat{c}_{j,\mathrm{pp}},\tilde{c}_{j,\mathrm{cp}})\right| \leqslant L^j_{\mathrm{tp}}|\widehat{c}_{j,\mathrm{tp}} - \tilde{c}_{j,\mathrm{tp}}| + L^j_{\mathrm{cp}}|\widehat{c}_{j,\mathrm{cp}} - \tilde{c}_{j,\mathrm{cp}}|. \quad (6.12)$$

Notice that $L^j_{\mathrm{pp}}|\widehat{c}_{j,\mathrm{pp}} - \widehat{c}_{j,\mathrm{pp}}|$ has not been included as it is zero since $\widehat{c}_{j,\mathrm{pp}} = \tilde{c}_{j,\mathrm{pp}}$. By taking expectation on both sides and applying Jensen's inequality to a concave function $x \mapsto \sqrt{x}$, we get:

$$\begin{aligned}&\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\left|\psi^j(\widehat{c}_{j,\mathrm{tp}},\widehat{c}_{j,\mathrm{pp}},\widehat{c}_{j,\mathrm{cp}}) - \psi^j(\tilde{c}_{j,\mathrm{tp}},\widehat{c}_{j,\mathrm{pp}},\tilde{c}_{j,\mathrm{cp}})\right|\right] \\ &\quad\leqslant L^j_{\mathrm{tp}}\sqrt{\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[(\widehat{c}_{j,\mathrm{tp}} - \tilde{c}_{j,\mathrm{tp}})^2\right]} + L^j_{\mathrm{cp}}\sqrt{\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[(\widehat{c}_{j,\mathrm{cp}} - \tilde{c}_{j,\mathrm{cp}})^2\right]}, \quad (6.13)\end{aligned}$$

Because $\tilde{c}_{j,\mathrm{tp}} = \mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}[\widehat{c}_{j,\mathrm{tp}}]$, we have

$$\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[(\widehat{c}_{j,\mathrm{tp}} - \tilde{c}_{j,\mathrm{tp}})^2\right] = \mathrm{Var}_{\mathbf{Y}|\mathbf{X}}(\widehat{c}_{j,\mathrm{tp}}) \leqslant \frac{1}{4n}, \quad (6.14)$$

as $\widehat{c}_{j,\mathrm{tp}} = \frac{1}{n}\sum_{i=1}^{n} y_{i,j}\widehat{y}_{i,j}$ is an average of n Bernoulli i.i.d. random variables $y_{i,j}\widehat{y}_{i,j}$, each having variance at most $\frac{1}{4}$. The same reasoning can be applied to $\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}[(\widehat{c}_{j,\mathrm{cp}} - \tilde{c}_{j,\mathrm{cp}})^2] \leqslant \frac{1}{4n}$. By applying these to (6.13), we conclude the

proof by getting:

$$\mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \mid \mathbf{X}]} \left[\left| \psi^j(\tilde{c}_{j,\mathrm{tp}}, \widehat{c}_{j,\mathrm{pp}}, \tilde{c}_{j,\mathrm{cp}}) - \psi^j(\widehat{c}_{j,\mathrm{tp}}, \widehat{c}_{j,\mathrm{pp}}, \widehat{c}_{j,\mathrm{cp}}) \right| \right] \leqslant \frac{L^j_{\mathrm{tp}} + L^j_{\mathrm{cp}}}{2\sqrt{n}} \,. \quad (6.15)$$

□

6.4 Special case of linear utilities

We start the discussion on optimization of semi-empirical (6.6) with a special case in which $\tilde{\Phi}_{\mathrm{ETU}}$ is linear in the prediction-dependent arguments, tp and pp, that is, if for all labels $j \in [m]$:

$$\psi^j(\mathrm{tp}, \mathrm{pp}, \mathrm{cp}) = f^{\mathrm{tp}}_j(\mathrm{cp}) \cdot \mathrm{tp} + f^{\mathrm{pp}}_j(\mathrm{cp}) \cdot \mathrm{pp} + f^{\mathrm{cp}}_j(\mathrm{cp}) \cdot \mathrm{cp} \,, \quad (6.16)$$

where both $f^{\mathrm{tp}}_j(\mathrm{cp})$ and $f^{\mathrm{pp}}_j(\mathrm{cp})$ do not depend on tp and pp, and $f^{\mathrm{cp}}_j(\mathrm{cp}) \cdot \mathrm{cp}$ is a constant and can be dropped in practice.

Aside from instance-wise weighted utilities (5.2), which are linear in all arguments, this form can be used for dependent on the label priors, e.g., power law weights of the form $f^{\mathrm{tp}}_j(\mathrm{cp}) = \mathrm{cp}^{-\beta}$ and $f^{\mathrm{pp}}_j(\mathrm{cp}) = f^{\mathrm{cp}}_j(\mathrm{cp}) = 0$, which reduce to macro-recall for $\beta = 1$. Another example is macro-balanced accuracy@k with $f^{\mathrm{tp}}_j(\mathrm{cp}) = \frac{1}{2\mathrm{cp}} + \frac{1}{2(1-\mathrm{cp})}$, $f^{\mathrm{pp}}_j(\mathrm{cp}) = \frac{-1}{2(1-\mathrm{cp})}$, and $f^{\mathrm{cp}}_j(\mathrm{cp}) = \frac{1}{2(1-\mathrm{cp})\mathrm{cp}} + \frac{-1}{2(1-\mathrm{cp})}$ For this type of metrics (6.16), one can approximate the value of cp for each label and turn the problem into optimization of instance-wise weighted utility (as in (5.2)), with gains:

$$g_{j,\mathrm{tp}} = f^{\mathrm{tp}}_j(\mathrm{cp}) + f^{\mathrm{pp}}_j(\mathrm{cp}) \,, \quad g_{j,\mathrm{fp}} = f^{\mathrm{pp}}_j(\mathrm{cp}) \,, \quad g_{j,\mathrm{fn}} = 0 \,, \quad g_{j,\mathrm{tn}} = 0 \,. \quad (6.17)$$

And hence the optimal prediction can be derived for each instance $\boldsymbol{x} \in \mathcal{X}$ separately, leading to:

$$\widehat{\boldsymbol{y}}^{\star}_i = \operatorname*{arg\,max}_{\widehat{\boldsymbol{y}} \in \mathcal{Y}^{@k}} \mathbb{E}_{\boldsymbol{y} \sim \mathbb{P}[\boldsymbol{y} \mid \boldsymbol{x}_i]} [\mathrm{U}@k(\boldsymbol{y}, \widehat{\boldsymbol{y}})] \quad (6.18)$$

for each $i \in [n]$. The optimal prediction of each $\widehat{\boldsymbol{y}}_i$ is then the optimal classifier for instance-wise utilities, which was already derived in Theorem 3.2.1:

$$\widehat{\boldsymbol{y}}^{\star}_i = \text{select-top-}k(\boldsymbol{a} \odot \boldsymbol{\eta}(\boldsymbol{x}_i) + \boldsymbol{b}) \,, \quad (6.19)$$

where:

$$\boldsymbol{a} = \boldsymbol{g}_{:,\mathrm{tn}} + \boldsymbol{g}_{:,\mathrm{tp}} - \boldsymbol{g}_{:,\mathrm{fp}} - \boldsymbol{g}_{:,\mathrm{fn}} \,, \qquad \boldsymbol{b} = \boldsymbol{g}_{:,\mathrm{fp}} - \boldsymbol{g}_{:,\mathrm{tn}} \,. \quad (6.20)$$

In this specific case, we end up with:

$$a_j = f^{\mathrm{tp}}_j(\mathrm{cp}) + f^{\mathrm{pp}}_j(\mathrm{cp}) - f^{\mathrm{pp}}_j(\mathrm{cp}) = f^{\mathrm{tp}}_j(\mathrm{cp}) \,, \qquad b_j = f^{\mathrm{pp}}_j(\mathrm{cp}) \,. \quad (6.21)$$

6.5 Block-coordinate ascent algorithm

For nonlinear utilities Ψ, even with the approximation of expected utility $\tilde{\Phi}_{\mathrm{ETU}}$, finding the optimal prediction can be a very hard discrete optimization problem in general. Taking into account the scale of XMLC problems, we propose to use an efficient approach based on block-coordinate ascent (BCA) that constructs a sequence of predictions with non-decreasing utility $\widehat{\mathbf{Y}}^0, \widehat{\mathbf{Y}}^1, \widehat{\mathbf{Y}}^2 \ldots$, so that we end up with a solution that is locally optimal.

Assume the predictions are fixed for all instances except $\boldsymbol{x}_s$, where they are given by $\boldsymbol{z}$. In that case, we can write the semi-empirical quantities from (6.5) as

$$\begin{aligned} \tilde{\boldsymbol{c}}_{:,\mathrm{tp}} &= \frac{1}{n}\left(\boldsymbol{\eta}(\boldsymbol{x}_s) \odot \boldsymbol{z} + \sum_{i \in [n] \setminus \{s\}} \boldsymbol{\eta}(\boldsymbol{x}_s) \odot \widehat{\boldsymbol{y}}_i\right), \\ \tilde{\boldsymbol{c}}_{:,\mathrm{pp}} &= \frac{1}{n}\left(\boldsymbol{z} + \sum_{i \in [n] \setminus \{s\}} \widehat{\boldsymbol{y}}_i\right). \end{aligned} \tag{6.22}$$

Plugging into (6.6) leads to the following optimization:

$$\max_{\boldsymbol{z} \in \mathcal{Y}^{@k}} \sum_{j=1}^{m} \psi^j \left(\frac{1}{n}\eta_j(\boldsymbol{x}_s) z_j + \frac{1}{n} \sum_{i \in [n] \setminus \{s\}} \eta_j(\boldsymbol{x}_i) \widehat{y}_{i,j},\ \frac{1}{n} z_j + \frac{1}{n} \sum_{i \in [n] \setminus \{s\}} \widehat{y}_{i,j},\ \frac{1}{n} \sum_{i=1}^{m} \eta_j(\boldsymbol{x}_i) \right). \tag{6.23}$$

As everything except $z \in \{0,1\}^m$ is given, we can interpret ψ^j as a linear function of z_j, $\psi^j(z_j)$, and define a gain vector with elements $g_j = \psi^j(1) - \psi^j(0)$. The optimal prediction $\boldsymbol{z}^\star$ is then given by $\boldsymbol{z}^\star = \text{select-top-}k(\boldsymbol{g})$, in a similar form as in the case of instance-wise metrics. We get $\widehat{\mathbf{Y}}^{t+1}$ by replacing the s-th row of $\widehat{\mathbf{Y}}^t$ with $\boldsymbol{z}^\star$, and know that $\tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}}^{t+1}) \geqslant \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}}^t)$. Then we switch to the next instance $s \leftarrow s + 1$ in the randomized order of instances (shuffle($[n]$)).

The algorithm starts with some initial prediction $\widehat{\mathbf{Y}}^0$, e.g., by selecting k random labels for each instance. Then it calculates the corresponding $\tilde{\mathbf{C}}^0$ according to (6.5). This confusion matrix can be cached and updated in each iteration to reduce the computational load. Each time a new prediction $\widehat{\mathbf{Y}}^{t+1}$ is calculated, the confusion matrix is updated as follows:

$$\begin{aligned} \tilde{\boldsymbol{c}}^{t+1}_{:,\mathrm{tp}} &:= \tilde{\boldsymbol{c}}^{t}_{:,\mathrm{tp}} + \frac{1}{n}\boldsymbol{\eta}(\boldsymbol{x}_s) \odot \left(\widehat{\boldsymbol{y}}^{t+1}_s - \widehat{\boldsymbol{y}}^{t}_s\right), \\ \widehat{\boldsymbol{c}}^{t+1}_{:,\mathrm{pp}} &:= \widehat{\boldsymbol{c}}^{t}_{:,\mathrm{pp}} + \frac{1}{n}\left(\widehat{\boldsymbol{y}}^{t+1}_s - \widehat{\boldsymbol{y}}^{t}_s\right). \end{aligned} \tag{6.24}$$

With this, we can compute (6.23) in $\mathcal{O}(m)$ time using the following formulas:

$$\begin{aligned} \psi^j(1) &:= \psi^j\left(\tilde{c}^t_{j,\mathrm{tp}} + \frac{1}{n}\eta_j(\boldsymbol{x}_s)(1 - \widehat{y}^t_{s,j}),\ \widehat{c}_{j,\mathrm{pp}} + \frac{1}{n}(1 - \widehat{y}^t_{s,j}),\ \tilde{c}_{j,\mathrm{cp}}\right), \\ \psi^j(0) &:= \psi^j\left(\tilde{c}^t_{j,\mathrm{tp}} - \frac{1}{n}\eta_j(\boldsymbol{x}_s)\widehat{y}^t_{s,j},\ \widehat{c}_{j,\mathrm{pp}} - \frac{1}{n}\widehat{y}^t_{s,j},\ \tilde{c}_{j,\mathrm{cp}}\right). \end{aligned} \tag{6.25}$$

The algorithm stops when the improvement in the objective value over a previous value is lower than ϵ after going over a whole set of instances. Because the BCA algorithm generates a sequence of predictions improving by at least ϵ in each iteration, this criterion ensures that the algorithm terminates. In practice, even with small ϵ, the algorithm usually terminates after a few iterations. We present the pseudocode of BCA in Algorithm 6.1.

Algorithm 6.1 BCA with semi-empirical ETU approximation

Require: a set of instances $\mathbf{X}^{n\times d}$, labels marginal probability estimator $\widehat{\boldsymbol{\eta}}$, label-wise utility function Ψ, required budget k, stopping condition criterion ϵ
Ensure: a prediction matrix $\widehat{\mathbf{Y}}^{n\times m}$ with $\widehat{\boldsymbol{y}}_i \in \{0,1\}^m$ with $\|\widehat{\boldsymbol{y}}\|_1 = k$ for all $i \in [n]$
1: initialize $\widehat{\mathbf{Y}}^{n\times m}$: $\widehat{\boldsymbol{y}}_i \leftarrow$ select-random-$k(m)$ for all $i \in [n]$
2: $\tilde{\boldsymbol{c}}_{:,\mathrm{tp}} \leftarrow \frac{1}{n}\sum_{i=1}^{m} \widehat{\boldsymbol{\eta}}(\boldsymbol{x}_i) \odot \widehat{\boldsymbol{y}}_i$
3: $\widehat{\boldsymbol{c}}_{:,\mathrm{pp}} \leftarrow \frac{1}{n}\sum_{i=1}^{m} \widehat{\boldsymbol{y}}_i$
4: $\tilde{\boldsymbol{c}}_{:,\mathrm{cp}} \leftarrow \frac{1}{n}\sum_{i=1}^{m} \widehat{\boldsymbol{\eta}}(\boldsymbol{x}_i)$
5: $u_{\mathrm{old}} \leftarrow -\infty$
6: $u_{\mathrm{new}} \leftarrow \Psi(\tilde{\boldsymbol{c}}_{:,\mathrm{tp}}, \widehat{\boldsymbol{c}}_{:,\mathrm{pp}}, \tilde{\boldsymbol{c}}_{:,\mathrm{cp}})$
7: **while** $u_{\mathrm{new}} > u_{\mathrm{old}} + \epsilon$ **do**
8: **for** $s \in$ shuffle($[n]$) **do**
9: $\tilde{\boldsymbol{c}}_{:,\mathrm{tp}} \leftarrow \tilde{\boldsymbol{c}}_{:,\mathrm{tp}} - \frac{1}{n}\widehat{\boldsymbol{\eta}}(\boldsymbol{x}_s) \odot \widehat{\boldsymbol{y}}_s$
10: $\widehat{\boldsymbol{c}}_{:,\mathrm{pp}} \leftarrow \widehat{\boldsymbol{c}}_{:,\mathrm{pp}} - \frac{1}{n}\widehat{\boldsymbol{y}}_s$
11: **for** $j \in [m]$ **do**
12: $\psi^j(1) \leftarrow \psi^j(\tilde{c}_{j,\mathrm{tp}} + \frac{1}{n}\widehat{\eta}_j(\boldsymbol{x}_s), \widehat{c}_{j,\mathrm{pp}} + \frac{1}{n}, \tilde{c}_{j,\mathrm{cp}})$
13: $\psi^j(0) \leftarrow \psi^j(\tilde{c}_{j,\mathrm{tp}}, \widehat{c}_{j,\mathrm{pp}}, \tilde{c}_{j,\mathrm{cp}})$
14: $g_j \leftarrow \psi^j(1) - \psi^j(0)$
15: $\widehat{\boldsymbol{y}}_s \leftarrow$ select-top-$k(\boldsymbol{g})$
16: $\tilde{\boldsymbol{c}}_{:,\mathrm{tp}} \leftarrow \tilde{\boldsymbol{c}}_{:,\mathrm{tp}} + \frac{1}{n}\widehat{\boldsymbol{\eta}}(\boldsymbol{x}_s) \odot \widehat{\boldsymbol{y}}_s$
17: $\widehat{\boldsymbol{c}}_{:,\mathrm{pp}} \leftarrow \widehat{\boldsymbol{c}}_{:,\mathrm{pp}} + \frac{1}{n}\widehat{\boldsymbol{y}}_s$
18: $u_{\mathrm{old}} \leftarrow u_{\mathrm{new}}$
19: $u_{\mathrm{new}} \leftarrow \Psi(\tilde{\boldsymbol{c}}_{:,\mathrm{tp}}, \widehat{\boldsymbol{c}}_{:,\mathrm{pp}}, \tilde{\boldsymbol{c}}_{:,\mathrm{cp}})$
20: **return** $\widehat{\mathbf{Y}}$

6.5.1 Computational complexity of the BCA algorithm

The time and space complexity of the single iteration over the test set of instances in Algorithm 6.1 are both $\mathcal{O}(nm + m)$. Since selection of top-k values in vector $\boldsymbol{g}$ can be done in linear time using the introspective selection algorithm [Musser, 1997], all the operations for a single instance s (lines 9-17) have complexity $\mathcal{O}(m)$. This complexity may still be too high for extreme classification problems, where both n and m are very large. Because of that, in Chapter 8, we show a more efficient variant of the BCA algorithm, which integrates well with efficient inference methods widely used in XMLC.

6.5.2 Global optimality for linear metrics

If Ψ is a linear function, corresponding to an instance-wise weighted utility (5.2), the BCA algorithm recovers the optimal solution in the first iteration, stopping

after the second. To show it, let us transform lines 11-14 of Algorithm 6.1:

$$\begin{aligned} g_j = \psi^j(1) - \psi^j(0) &= \psi^j(\mathrm{tp} + \widehat{\eta}_j(\boldsymbol{x}_s), \mathrm{pp} + 1, \mathrm{cp}) - \psi^j(\mathrm{tp}, \mathrm{pp}, \mathrm{cp}) \\ &= (\mathrm{tp} + \widehat{\eta}_j(\boldsymbol{x}_s) - \mathrm{tp}) f_{\mathrm{tp}}(\mathrm{cp}) + (\mathrm{pp} + 1 - \mathrm{pp}) f_{\mathrm{pp}}(\mathrm{cp}) \\ &= \widehat{\eta}_j(\boldsymbol{x}_s) f_{\mathrm{tp}}(\mathrm{cp}) + f_{\mathrm{pp}}(\mathrm{cp}) \,. \end{aligned} \tag{6.26}$$

One can observe that the dependence on other instances cancels out. This leads exactly to the solution presented in Section 6.4, meaning that for each instance, the block coordinate ascent algorithm will select the optimal decision in terms of $\tilde{\Phi}_{\mathrm{ETU}}$, i.e., semi-empirical ETU (6.6). As discussed it earlier, it means that for fully linear metrics this will be the Bayes-optimal decision, otherwise it is approximately optimal in the sense of Theorem 6.3.2. In both cases, the algorithm will perform a second pass over the entire dataset in order to check the stopping criterion. This second pass will leave all predictions unchanged.

> **BCA for metrics without a budget constraint**
>
> The introduced BCA algorithm can be used for the optimization of metrics without the budget k constraint. The general idea behind the algorithm is to choose prediction $\widehat{\boldsymbol{y}}_s$ for an instance $\boldsymbol{x}_s$, such that it maximizes the expected utility gain for this instance (6.23), with predictions for all the other instances in the set as fixed. The modification simply involves changing the prediction rule from selecting top k labels with the highest values of expected gain g_j (line 15 of Algorithm 6.1) to select all labels with expected gain larger than 0.

6.6 Regret bound under LPE misspecification

The actual implementation cannot obviously use the unknown values $\boldsymbol{\eta}(\boldsymbol{x})$, but instead has to rely on the estimates $\widehat{\boldsymbol{\eta}}(\boldsymbol{x})$ when computing the predictions. Because of that, as before, we are interested in quantifying the regret that the classifier suffers in this case.

Given a decomposable metric of the form (6.2), which is invariant under instance reordering, let us define the output of optimal (Bayes) classifier $\boldsymbol{h}^\star(\mathbf{X}) = \mathbf{Y}^\star$ as the one which maximizes the expected utility with respect to the true label probabilities.

$$\mathbf{Y}^\star \in \operatorname*{arg\,max}_{\widehat{\mathbf{Y}} \in \mathcal{Y}^{@k,n}} \underbrace{\mathbb{E}_{\mathbf{Y} \sim \boldsymbol{\eta}(\mathbf{X})} \left[\Psi(\widehat{\boldsymbol{c}}_{:,\mathrm{tp}}, \widehat{\boldsymbol{c}}_{:,\mathrm{pp}}, \widehat{\boldsymbol{c}}_{:,\mathrm{cp}}) \right]}_{\Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}})} . \tag{6.27}$$

We use $\mathbf{Y} \sim \boldsymbol{\eta}(\mathbf{X})$ notation here to emphasize the dependency of Φ_{ETU} on the label marginal probabilities. Then, let $\tilde{\mathbf{Y}}^\dagger$ be the plug-in prediction matrix optimizing the semi-empirical ETU with plugged-in probability estimates:

$$\tilde{\mathbf{Y}}^\dagger \in \operatorname*{arg\,max}_{\widehat{\mathbf{Y}} \in \mathcal{Y}^{@k,n}} \underbrace{\Psi(\mathbb{E}_{\mathbf{Y} \sim \widehat{\boldsymbol{\eta}}(\mathbf{X})}[\widehat{\boldsymbol{c}}_{:,\mathrm{tp}}], \widehat{\boldsymbol{c}}_{:,\mathrm{pp}}, \mathbb{E}_{\mathbf{Y} \sim \widehat{\boldsymbol{\eta}}(\mathbf{X})}[\widehat{\boldsymbol{c}}_{:,\mathrm{cp}}])}_{\tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}})} . \tag{6.28}$$

Theorem 6.6.1. *Let $\tilde{\mathbf{Y}}^\dagger$ be defined as above. Under the assumptions of Theorem 6.3.2:*

$$\begin{aligned}\mathrm{Reg}_{\mathrm{ETU}}\left(\tilde{\mathbf{Y}}^\dagger;\mathbf{X}\right) &= \Phi_{\mathrm{ETU}}\left(\mathbf{Y}^\star;\mathbf{X}\right) - \Phi_{\mathrm{ETU}}\left(\tilde{\mathbf{Y}}^\dagger;\mathbf{X}\right) \\ &\leqslant \frac{m}{\sqrt{n}}B + 2\frac{\sqrt{m}}{n}B\sum_{i=1}^{n}\left\|\boldsymbol{\eta}(\boldsymbol{x}_i) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x}_i)\right\|_2, \end{aligned} \tag{6.29}$$

where $B \coloneqq \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(L^j_{\mathrm{tp}}(\tilde{c}_{j,\mathrm{cp}}) + L^j_{\mathrm{cp}}(\widehat{c}_{j,\mathrm{cp}})\right)^2}$ is the quadratic mean of the Lipschitz constants.

Proof (sketch, the full proof can be found in Appendix A.3.4). Due to the length of the proof, we present here only the general idea behind it. We first show that for metrics with cp-Lipschitz components, the Ψ-regret of the resulting semi-ETU predictor (A.48), which is the suboptimality of $\tilde{\mathbf{Y}}^\dagger$ (with respect to $\mathbf{Y}^\star$) in terms of Φ_{ETU}, is well-controlled and upper-bounded by the L_1 estimation error of the marginals. As the resulting expression is unwieldy, we then apply some further bounding to arrive at the much simpler result stated above, which is expressed in terms of L_2 estimation error. A similar statement, presented in Appendix A.3.5, can be made for the unapproximated ETU case.

Based on the above theorem, we can conclude that the methods described in Sections 6.3 to 6.5 can be used with probability estimates replacing the true marginals. As long as the estimator is reliable, the resulting predictions will have small regret. □

6.7 The case of coverage@k

One measure for which the introduced ETU approximation (6.6) is not exact is coverage,

$$\psi_{\mathrm{Cov}}(\mathrm{tp},\mathrm{pp},\mathrm{cp}) \coloneqq \mathbb{1}\left[\mathrm{tp} > 0\right], \tag{6.30}$$

as it is not only nonlinear but also not cp-Lipschitz as it is not continuous. Therefore, the bounds from Theorems 6.3.2 and 6.6.1 do not apply to it. Nevertheless, it is a popular auxiliary measure in XMLC. Remark that optimization of coverage@k makes sense in the ETU setting only, as a fully random classifier in the PU setting is optimal as it obtains coverage@k of 1 at the population level. Because of that,

we consider it separately in this section and derive a formula for its Φ_{ETU}:

$$\begin{aligned}
\Phi_{\text{ETU}}(\boldsymbol{h}) &= \mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \mid \mathbf{X}]} \left[\Psi_{\text{Cov}}(\widehat{\mathbf{C}}(\mathbf{Y}, \boldsymbol{h}(\mathbf{X})) \right] = \mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \mid \mathbf{X}]} \left[\Psi_{\text{Cov}}(\widehat{\mathbf{C}}(\mathbf{Y}, \widehat{\mathbf{Y}}) \right] \\
&= \mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \mid \mathbf{X}]} \left[\frac{1}{m} \sum_{j=1}^{m} \mathbb{1} \left[\widehat{c}_{j,\text{tp}} > 0 \right] \right] \\
&= \mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \mid \mathbf{X}]} \left[\frac{1}{m} \sum_{j=1}^{m} \left(1 - \mathbb{1} \left[\widehat{c}_{j,\text{tp}} = 0 \right] \right) \right] \\
&= \mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \mid \mathbf{X}]} \left[\frac{1}{m} \sum_{j=1}^{m} \left(1 - \prod_{i=1}^{n} (1 - y_{i,j} \widehat{y}_{i,j}) \right) \right] \\
&= \mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \mid \mathbf{X}]} \left[1 - \frac{1}{m} \sum_{j=1}^{m} \prod_{i=1}^{n} (1 - y_{i,j} \widehat{y}_{i,j}) \right] . \qquad (6.31)
\end{aligned}$$

Because we assume labels for one instance to be independent on all other instances, i.e., $\mathbb{P}[\mathbf{Y}|\mathbf{X}] = \prod_{i=1}^{n} \mathbb{P}[\boldsymbol{y}_i|\boldsymbol{x}_i]$, we obtain:

$$\begin{aligned}
\Phi_{\text{ETU}}(\boldsymbol{h}) &= 1 - \frac{1}{m} \sum_{j=1}^{m} \prod_{i=1}^{n} \left(1 - \mathbb{E}_{\mathbf{Y} \sim \mathbb{P}[\mathbf{Y} \mid \mathbf{X}]} [y_{i,j} \widehat{y}_{i,j}] \right) \\
&= 1 - \frac{1}{m} \sum_{j=1}^{m} \prod_{i=1}^{n} (1 - \mathbb{P}[y_{i,j} = 1 | \boldsymbol{x}_i] \widehat{y}_{i,j}) \\
&= 1 - \frac{1}{m} \sum_{j=1}^{m} \prod_{i=1}^{n} (1 - \eta_j(\boldsymbol{x}_i) \widehat{y}_{i,j}) \ . \qquad (6.32)
\end{aligned}$$

Based on this result, we can construct a block coordinate ascent procedure which, for predictions being fixed for all instances except $\boldsymbol{x}_s$, optimizes the following problem:

$$\max_{\boldsymbol{z} \in \mathcal{Y}^{@k}} \sum_{j=1}^{m} \psi_{\text{Cov}}^{j}(z_j) = \sum_{j=1}^{m} \left(1 - (1 - \eta_j(\boldsymbol{x}_i) z_j) \prod_{i \in [n] \setminus \{s\}} (1 - \eta_j(\boldsymbol{x}_i) \widehat{y}_{i,j}) \right) . \qquad (6.33)$$

Analogously to Section 6.5, everything except $z_j \in \{0, 1\}$ is given and we can define a gain vector with elements $g_j = \psi_{\text{Cov}}^{j}(1) - \psi_{\text{Cov}}^{j}(0)$. Once again, the optimal prediction is given by $\boldsymbol{z}^{\star} = \text{select-top-}k(\boldsymbol{g})$, and we get $\widehat{\mathbf{Y}}^{t+1}$ by replacing the s-th row of $\widehat{\mathbf{Y}}^{t}$ with $\boldsymbol{z}^{\star}$, resulting with $\tilde{\Phi}_{\text{ETU}}(\widehat{\mathbf{Y}}^{t+1}) \geqslant \tilde{\Phi}_{\text{ETU}}(\widehat{\mathbf{Y}}^{t})$. Then we switch to the next instance $s \leftarrow s + 1$ and repeat utill the stopping criterion is met.

Notice that $\prod_{i=1}^{n} (1 - \eta_j(\boldsymbol{x}_i) \widehat{y}_{i,j})$ corresponds to the probability of label j being irrelevant for all instances for which it was selected. We denote this value as f_j and use it to speed up computations in each iteration of the algorithm:

$$f_j^0 \coloneqq \prod_{i=1}^{n} \left(1 - \eta_j(\boldsymbol{x}_i) \widehat{y}_{i,j}^0 \right) , \qquad f_j^{t+1} \coloneqq f_j^t \frac{1 - \eta_j(\boldsymbol{x}_s) \widehat{y}_{s,j}^{t+1}}{1 - \eta_j(\boldsymbol{x}_s) \widehat{y}_{s,j}^{t}} . \qquad (6.34)$$

We can then compute the gain vector using the following formula:

$$\begin{aligned} g_j &= \psi^j_{\text{Cov}}(1) - \psi^j_{\text{Cov}}(0) \\ &= \left(1 - f_j^t \frac{1-\eta_j(\boldsymbol{x}_s)}{1-\eta_j(\boldsymbol{x}_s)\widehat{y}^t_{s,j}}\right) - \left(1 - f_j^t \frac{1}{1-\eta_j(\boldsymbol{x}_s)\widehat{y}^t_{s,j}}\right) \\ &= \frac{\eta_j(\boldsymbol{x}_s) f_j^t}{1-\eta_j(\boldsymbol{x}_s)\widehat{y}^t_{s,j}} \,. \end{aligned} \tag{6.35}$$

This block coordinate ascent procedure for coverage is presented as Algorithm 6.2.

Algorithm 6.2 BCA for coverage@k

Require: a set of instances $\mathbf{X}^{n\times d}$, labels marginal probability estimator $\widehat{\boldsymbol{\eta}}$, required budget k, stopping condition criterion ϵ
Ensure: a prediction matrix $\widehat{\mathbf{Y}}^{n\times m}$ with $\widehat{\boldsymbol{y}}_i \in \{0,1\}^m$ with $\|\widehat{\boldsymbol{y}}\|_1 = k$ for all $i \in [n]$

initialize $\widehat{\mathbf{Y}}^{n\times m}$: $\widehat{\boldsymbol{y}}_i \leftarrow$ select-random-$k(m)$ for all $i \in [n]$
$\boldsymbol{f} \leftarrow \prod_{i=1}^n (\mathbf{1} - \widehat{\boldsymbol{\eta}}(\boldsymbol{x}_i) \odot \widehat{\boldsymbol{y}}_i)$
$u_{\text{old}} \leftarrow -\infty$
$u_{\text{new}} \leftarrow 1 - \frac{1}{m}\sum_{j=1}^m f_j$
while $u_{\text{new}} > u_{\text{old}} + \epsilon$ **do**
  for $s \in$ shuffle($[n]$) **do**
    $\boldsymbol{f} \leftarrow \boldsymbol{f} \odot (\mathbf{1} - \widehat{\boldsymbol{\eta}}(\boldsymbol{x}_s) \odot \widehat{\boldsymbol{y}}_s)^{-1}$
    $\boldsymbol{g} \leftarrow \boldsymbol{f} \odot \widehat{\boldsymbol{\eta}}(\boldsymbol{x}_s)$
    $\widehat{\boldsymbol{y}}_s \leftarrow$ select-top-$k(\boldsymbol{g})$
    $\boldsymbol{f} \leftarrow \boldsymbol{f} \odot (\mathbf{1} - \widehat{\boldsymbol{\eta}}(\boldsymbol{x}_s) \odot \widehat{\boldsymbol{y}}_s)$
  $u_{\text{old}} \leftarrow u_{\text{new}}$
  $u_{\text{new}} \leftarrow 1 - \frac{1}{m}\sum_{j=1}^m f_j$
return $\widehat{\mathbf{Y}}$

Because the bound from Theorem 6.6.1 does not apply to the regret of the classifier under inaccurate estimates of $\boldsymbol{\eta}$. We derive a new bound using the (6.32). Similarly to Section 6.6, let us define the optimal prediction matrix $\mathbf{Y}^\star$ that optimizes the expected utility with respect to the true label probabilities:

$$\mathbf{Y}^\star \in \operatorname*{arg\,max}_{\widehat{\mathbf{Y}} \in \mathcal{Y}^{@k,n}} \mathbb{E}_{\mathbf{Y}\sim\boldsymbol{\eta}(\mathbf{X})} \left[\Psi_{\text{Cov}}\left(\widehat{\mathbf{C}}\left(\mathbf{Y}, \widehat{\mathbf{Y}}\right)\right)\right] . \tag{6.36}$$

Then, let $\mathbf{Y}^\dagger$ be the prediction matrix optimizing the expected utility with respect to the estimated label probabilities:

$$\mathbf{Y}^\dagger \in \operatorname*{arg\,max}_{\widehat{\mathbf{Y}} \in \mathcal{Y}^{@k,n}} \mathbb{E}_{\mathbf{Y}\sim\widehat{\boldsymbol{\eta}}(\mathbf{X})} \left[\Psi_{\text{Cov}}\left(\widehat{\mathbf{C}}\left(\mathbf{Y}, \widehat{\mathbf{Y}}\right)\right)\right] . \tag{6.37}$$

Theorem 6.7.1. *Let $\mathbf{Y}^\star$ and $\mathbf{Y}^\dagger$ be defined as above, then:*

$$\text{Reg}_{\text{ETU}}\left(\mathbf{Y}^\dagger; \mathbf{X}\right) = \Phi_{\text{ETU}}\left(\mathbf{Y}^\star; \mathbf{X}\right) - \Phi_{\text{ETU}}\left(\mathbf{Y}^\dagger; \mathbf{X}\right) \leqslant \sum_{i=1}^n 2k \max_{j\in[m]} |\eta_j(\boldsymbol{x}_i) - \widehat{\eta}_j(\boldsymbol{x}_i)| \,. \tag{6.38}$$

Proof.

$$\begin{aligned}
\mathrm{Reg}_{\mathrm{ETU}}\left(\mathbf{Y}^{\dagger};\mathbf{X}\right) &= \Phi_{\mathrm{ETU}}\left(\mathbf{Y}^{\star};\mathbf{X}\right) - \Phi_{\mathrm{ETU}}\left(\mathbf{Y}^{\dagger};\mathbf{X}\right) \\
&= 1 - \frac{1}{m}\sum_{j=1}^{m}\prod_{i=1}^{n}\left(1-\eta_j(\boldsymbol{x}_i)y_{i,j}^{\star}\right) - \left(1 - \frac{1}{m}\sum_{j=1}^{m}\prod_{i=1}^{n}\left(1-\eta_j(\boldsymbol{x}_i)y_{i,j}^{\dagger}\right)\right) \\
&= \frac{1}{m}\sum_{j=1}^{m}\prod_{i=1}^{n}\left(1-\eta_j(\boldsymbol{x}_i)y_{i,j}^{\dagger}\right) - \prod_{i=1}^{n}\left(1-\eta_j(\boldsymbol{x}_i)y_{i,j}^{\star}\right) . \qquad (6.39)
\end{aligned}$$

Now we add and subtract the following two terms inside the sum $\sum_{j=1}^{m}$:

$$\prod_{i=1}^{n}\left(1-\widehat{\boldsymbol{\eta}}_j(\boldsymbol{x}_i)y_{i,j}^{\dagger}\right), \qquad \prod_{i=1}^{n}\left(1-\widehat{\boldsymbol{\eta}}_j(\boldsymbol{x}_i)y_{i,j}^{\star}\right), \qquad (6.40)$$

then reorganize and bound the expression using the $\prod_{i=1}^{n} a_i - \prod_{i=1}^{n} b_i \leqslant \sum_{i=1}^{n} a_i - b_i$ (for $a, b \in [0,1]$) inequality:

$$\begin{aligned}
\mathrm{Reg}_{\mathrm{ETU}}\left(\mathbf{Y}^{\dagger};\mathbf{X}\right) = \frac{1}{m}\sum_{j=1}^{m}\Bigg(&\underbrace{\prod_{i=1}^{n}\left(1-\eta_j(\boldsymbol{x}_i)y_{i,j}^{\dagger}\right) - \prod_{i=1}^{n}\left(1-\widehat{\boldsymbol{\eta}}_j(\boldsymbol{x}_i)y_{i,j}^{\dagger}\right)}_{\leqslant\sum_{i=1}^{n}|\eta_j(\boldsymbol{x}_i)-\widehat{\boldsymbol{\eta}}_j(\boldsymbol{x}_i)|y_{i,j}^{\dagger}} \\
&+\underbrace{\prod_{i=1}^{n}\left(1-\widehat{\boldsymbol{\eta}}_j(\boldsymbol{x}_i)y_{i,j}^{\star}\right) - \prod_{i=1}^{n}\left(1-\eta_j(\boldsymbol{x}_i)y_{i,j}^{\star}\right)}_{\leqslant\sum_{i=1}^{n}|\eta_j(\boldsymbol{x}_i)-\widehat{\boldsymbol{\eta}}_j(\boldsymbol{x}_i)|y_{i,j}^{\star}} \\
&+\underbrace{\prod_{i=1}^{n}\left(1-\widehat{\boldsymbol{\eta}}_j(\boldsymbol{x}_i)y_{i,j}^{\dagger}\right) - \prod_{i=1}^{n}\left(1-\widehat{\boldsymbol{\eta}}_j(\boldsymbol{x}_i)y_{i,j}^{\star}\right)}_{\leqslant 0}\Bigg) \\
\leqslant \frac{1}{m}\sum_{j=1}^{m}&\left(\sum_{i=1}^{n}\left|\eta_j(\boldsymbol{x}_i)-\widehat{\eta}_j(\boldsymbol{x}_i)\right|y_{i,j}^{\dagger} + \sum_{i=1}^{n}\left|\eta_j(\boldsymbol{x}_i)-\widehat{\eta}_j(\boldsymbol{x}_i)\right|y_{i,j}^{\star}\right). \qquad (6.41)
\end{aligned}$$

Next, we bound each L_1 error, $|\eta_j(\boldsymbol{x}_i) - \widehat{\eta}_j(\boldsymbol{x}_i)|$ by $\max_{j\in[m]}|\eta_j(\boldsymbol{x}_i) - \widehat{\eta}_j(\boldsymbol{x}_i)|$. There are at most $\|\boldsymbol{y}_i^{@k,\star} \vee \boldsymbol{y}_i^{@k,\dagger}\|_1 \leqslant 2k$ such terms that stay positive for each instance i. Therefore:

$$\begin{aligned}
\mathrm{Reg}_{\mathrm{ETU}}\left(\mathbf{Y}^{\dagger};\mathbf{X}\right) &\leqslant \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{n} 2k \max_{j\in[m]}\left|\eta_j(\boldsymbol{x}_i)-\widehat{\eta}_j(\boldsymbol{x}_i)\right| \\
&\leqslant \sum_{i=1}^{n} 2k \max_{j\in[m]}\left|\eta_j(\boldsymbol{x}_i)-\widehat{\eta}_j(\boldsymbol{x}_i)\right| . \qquad (6.42)
\end{aligned}$$

□

Interestingly, the bound is very similar to the bound we obtained for general instance-wise weighted utilities@k in Section 3.2.

6.8 Greedy optimization of semi-empirical ETU objective

The inference strategy presented in Algorithm 6.1 requires multiple passes over the entire dataset. This might be computationally expensive for large datasets and requires the whole dataset to be available at once. We thus propose a greedy version of this algorithm, which performs just a single pass over the dataset, taking into account only the statistics (expected confusion matrix) of previously seen instances for which it has already decided on the prediction. This way, the greedy algorithm adjusts the proposed approach to the semi-online setting where the whole dataset cannot be observed at once. We present the pseudocode of the greedy algorithm in Algorithm 6.3 and its specialized version for coverage in Algorithm 6.4.

Algorithm 6.3 Greedy

Require: a set of instances $\mathbf{X}^{n\times d}$, labels marginal probability estimator $\widehat{\boldsymbol{\eta}}$, required budget k, stopping condition criterion ϵ
Ensure: a prediction matrix $\widehat{\mathbf{Y}}^{n\times m}$ with $\widehat{\boldsymbol{y}}_i \in \{0,1\}^m$ with $\|\widehat{\boldsymbol{y}}\|_1 = k$ for all $i \in [n]$

$\tilde{\boldsymbol{c}}_{:,\mathrm{tp}} \leftarrow \mathbf{0}$
$\widehat{\boldsymbol{c}}_{:,\mathrm{pp}} \leftarrow \mathbf{0}$
$\tilde{\boldsymbol{c}}_{:,\mathrm{cp}} \leftarrow \mathbf{0}$
for $i \in [n]$ **do**
  $\tilde{\boldsymbol{c}}_{:,\mathrm{cp}} \leftarrow \tilde{\boldsymbol{c}}_{:,\mathrm{cp}} + \frac{1}{n}\widehat{\boldsymbol{\eta}}(\boldsymbol{x}_i)$
  for $j \in [m]$ **do**
    $\psi^j(1) \leftarrow \psi^j(\tilde{c}_{j,\mathrm{tp}} + \frac{1}{n}\widehat{\eta}(\boldsymbol{x}_s), \widehat{c}_{j,\mathrm{pp}} + \frac{1}{n}, \tilde{c}_{j,\mathrm{cp}})$
    $\psi^j(0) \leftarrow \psi^j(\tilde{c}_{j,\mathrm{tp}}, \widehat{c}_{j,\mathrm{pp}}, \tilde{c}_{j,\mathrm{cp}})$
    $g_j \leftarrow \psi^j(1) - \psi^j(0)$
  $\widehat{\boldsymbol{y}}_i \leftarrow$ select-top-$k(\boldsymbol{g})$
  $\tilde{\boldsymbol{c}}_{:,\mathrm{tp}} \leftarrow \tilde{\boldsymbol{c}}_{:,\mathrm{tp}} + \frac{1}{n}\widehat{\boldsymbol{\eta}}(\boldsymbol{x}_i) \odot \widehat{\boldsymbol{y}}_i$
  $\widehat{\boldsymbol{c}}_{:,\mathrm{pp}} \leftarrow \widehat{\boldsymbol{c}}_{:,\mathrm{pp}} + \frac{1}{n}\widehat{\boldsymbol{y}}_i$
return $\widehat{\mathbf{Y}}$

Algorithm 6.4 Greedy for coverage@k

Require: a set of instances $\mathbf{X}^{n\times d}$, labels marginal probability estimator $\widehat{\boldsymbol{\eta}}$, required budget k, stopping condition criterion ϵ
Ensure: a prediction matrix $\widehat{\mathbf{Y}}^{n\times m}$ with $\widehat{\boldsymbol{y}}_i \in \{0,1\}^m$ with $\|\widehat{\boldsymbol{y}}\|_1 = k$ for all $i \in [n]$

$\boldsymbol{f} \leftarrow \mathbf{1}$
for $i \in [n]$ **do**
  $\boldsymbol{g} \leftarrow \boldsymbol{f} \odot \widehat{\boldsymbol{\eta}}(\boldsymbol{x}_i)$
  $\widehat{\boldsymbol{y}}_i \leftarrow$ select-top-$k(\boldsymbol{g})$
  $\boldsymbol{f} \leftarrow \boldsymbol{f} \odot (\mathbf{1} - \widehat{\boldsymbol{\eta}}(\boldsymbol{x}_i) \odot \widehat{\boldsymbol{y}}_i)$
return $\widehat{\mathbf{Y}}$

6.9 Summary of the chapter

In this chapter, we have analyzed the problem of optimization of label-wise utilities budgeted at k, i.e., utilities that decompose linearly over labels in the expected test utility (ETU) framework. We demonstrated that the expected utility for label-wise utilities is determined by the marginal conditional probability of labels $\eta_j(\boldsymbol{x}) = \mathbb{P}[y_j = 1 \,|\, \boldsymbol{x}]$, making optimization of these measures compatible with the current approaches to XMLC. However, calculating the expected utility requires averaging over all combinations of possible confusion matrices, which is computationally intractable with a large number of instances and labels. Because of that, we have proposed a computationally efficient way of approximating the expected utility and block coordinate ascent (BCA) algorithm for its optimization. For the proposed approach, we have demonstrated regret guarantees and robustness against label probability estimator (LPE) misspecification, expressed by the L_2 estimation error of $\eta_j(\boldsymbol{x})$.

Overall, we have identified four categories of utilities in the ETU framework that differ in the complexity of the optimization algorithm – whether to use instance-wise optimization (Section 6.4) or the block coordinate ascent (Section 6.5) – and the guarantees for the result – whether semi-empirical quantities (Section 6.3) lead to an optimal solution or a suboptimal one with an error bounded by Theorem 6.3.2:

1. **Fully linear**: Optimal predictions for metrics that are linear in all entries of the confusion matrix (in $\mathrm{tp}, \mathrm{pp}, \mathrm{cp}$), as in (5.2), can be solved exactly in an instance-wise manner. Examples are classical metrics such as instance-wise precision@k, propensity-scored precision@k, or Hamming-loss@k.
2. **Linear in predictions**: Approximately optimal predictions for metrics that are linear in the predictions (in tp, pp) as given in (6.16) can be obtained using instance-wise optimization, by switching from Φ_{ETU} to $\tilde{\Phi}_{\mathrm{ETU}}$. An example is macro-recall@k and macro-balanced accuracy@k.
3. **Linear in labels**: If a metric is linear in the label variables (in tp, cp) as given in (6.8), then $\tilde{\Phi}_{\mathrm{ETU}} \equiv \Phi_{\mathrm{ETU}}$. However, the resulting combinatorial optimization problem for $\tilde{\Phi}_{\mathrm{ETU}}$ is still complex enough, and we can solve it only locally. An example is macro-precision@k.
4. **Nonlinear metrics**: If none of the above apply, we have $\tilde{\Phi}_{\mathrm{ETU}} \neq \Phi_{\mathrm{ETU}}$, and have to solve it locally using block-coordinate ascent (BCA). This is the case of macro-F_β-measure@k or macro-Jaccard similarity@k. Coverage@k is also a nonlinear metric, but as we showed in Section 6.7, we can efficiently calculate Φ_{ETU} for it, however, we still can only solve it locally using a specialized variant of BCA procedure.

We present the results of the BCA combined with a popular label probability estimator for extreme multi-label classification problems in Chapter 9, demonstrating that it is an effective and efficient method for optimizing label-wise utilities in the ETU framework.

7

Optimization of label-wise metrics at k under population utility framework

Following the ETU framework, in this chapter, we move to the problem of optimizing label-wise metrics at k under the population utility (PU) framework, and propose an efficient algorithm for this task. This chapter mostly summarizes the results co-authored in Schultheis et al. [2024].

7.1 Randomized classifier and expected confusion matrix

We start our analysis by defining $\boldsymbol{h}^{\text{rnd@}k} \in \mathcal{H}^{@k}$, which is a randomized classifier budgeted at k. Such a classifier $\boldsymbol{h}^{\text{rnd@}k}$ can be defined on a vector of marginal probabilities of sampling each label $\boldsymbol{\theta} \in [0,1]^m$, such that $\|\boldsymbol{\theta}\|_1 = k$, since one can construct a distribution over binary vectors $\widehat{\boldsymbol{y}}$ with $\|\widehat{\boldsymbol{y}}\|_1 = k$ and marginals $\theta_j = \mathbb{P}\left[\widehat{y}_j = 1 \,|\, \boldsymbol{x}\right]$ for all labels j. This can be accomplished using, e.g., Madow's sampling scheme, which we present in detail in Appendix A.4.1. Then, let us define the space $\Delta^{@k} := \{\boldsymbol{\theta} \in [0,1]^m \colon \|\boldsymbol{\theta}\|_1 = k\}$, where $\boldsymbol{\theta}(\boldsymbol{x})$ are all (measurable) functions of the form $\boldsymbol{\theta} \colon \mathcal{X} \to \Delta^{@k}$. We denote the set of such functions as $\mathcal{S}^{@k}$, and define a randomized classifier as:

$$\boldsymbol{h}^{\text{rnd@}k}(\boldsymbol{x}) := \text{sample-}k\left(\boldsymbol{\theta}(\boldsymbol{x})\right) , \tag{7.1}$$

where sample-k is a function that samples a k-hot encoded vector according to marginal distribution $\boldsymbol{\theta}(\boldsymbol{x})$ (e.g., using Madow's sampling).

As a reminder, in the PU framework, we are interested in the utility of a classifier calculated over the expected value of a confusion matrix calculated on the population distribution $\mathbb{P}\left[\boldsymbol{x}, \boldsymbol{y}\right]$:

$$\Phi_{\text{PU}}(\boldsymbol{h}) := \Psi\left(\mathbb{E}_{\boldsymbol{x},\boldsymbol{y} \sim \mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}\left[\widehat{\mathbf{C}}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x}))\right]\right) . \tag{7.2}$$

Using the definition of randomized classifier $\boldsymbol{h}^{\text{rnd@}k}$, we define the expected multi-

label confusion matrix of a randomized classifier:

$$\mathbf{C}(\boldsymbol{h}^{\mathrm{rnd@}k}) := \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}\left[\widehat{\mathbf{C}}(\boldsymbol{y},\boldsymbol{h}^{\mathrm{rnd@}k}(\boldsymbol{x}))\right], \tag{7.3}$$

where $\widehat{\mathbf{C}}(\boldsymbol{y},\boldsymbol{h}^{\mathrm{rnd@}k}(\boldsymbol{x}))$ is the confusion matrix, as defined in (2.5). This results in $\mathbf{C}(\boldsymbol{h}^{\mathrm{rnd@}k}) = [\boldsymbol{c}(h_1^{\mathrm{rnd@}k}),\ldots,\boldsymbol{c}(h_m^{\mathrm{rnd@}k})]$ where

$$\boldsymbol{c}(h_j^{\mathrm{rnd@}k}) = [\mathrm{tn}(h_j^{\mathrm{rnd@}k}),\mathrm{fp}(h_j^{\mathrm{rnd@}k}),\mathrm{fn}(h_j^{\mathrm{rnd@}k}),\mathrm{tp}(h_j^{\mathrm{rnd@}k})], \tag{7.4}$$

is the expected binary confusion matrix with entries:

$$\begin{aligned}
\mathrm{tn}(h_j^{\mathrm{rnd@}k}) &:= \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}[(1-y_j)(1-h_j^{\mathrm{rnd@}k}(\boldsymbol{x})] \\
&= \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}[(1-\eta_j(\boldsymbol{x}))(1-\theta_j(\boldsymbol{x}))] \\
\mathrm{fp}(h_j^{\mathrm{rnd@}k}) &:= \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}[(1-\eta_j(\boldsymbol{x}))\theta_j(\boldsymbol{x})] \\
\mathrm{fn}(h_j^{\mathrm{rnd@}k}) &:= \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}[\eta_j(\boldsymbol{x})(1-\theta_j(\boldsymbol{x}))] \\
\mathrm{tp}(h_j^{\mathrm{rnd@}k}) &:= \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}[\eta_j(\boldsymbol{x})\theta_j(\boldsymbol{x})]\,.
\end{aligned} \tag{7.5}$$

The set of all possible expected binary confusion matrices is the same as the set of all possible empirical binary confusion matrices $\mathcal{C} = \left\{\boldsymbol{c}\in[0,1]^4 : \|\boldsymbol{c}\|_1 = 1\right\}$, as defined in Section 2.2.1. We use it here to define the set of possible expected multi-label confusion matrices of a classifier budgeted at k

$$\mathcal{C}^{m,@k} = \left\{\mathbf{C}\in[0,1]^{m,4} : \sum_{j=1}^{m}\widehat{c}_{j,\mathrm{fp}} + \widehat{c}_{j,\mathrm{tp}} = k\right\}. \tag{7.6}$$

Because the PU framework requires the metric to be defined on a confusion matrix, we will mostly use the notation of utility as a function of the confusion matrix $\Psi(\mathbf{C})$ in this chapter.

While in general, given two confusion matrices, we cannot say which one is better than another without knowing the utility function, however, we can impose a partial order that most reasonable performance metrics should respect. We assume the utility should increase or stay the same if both true positives and true negatives increase. The below definition is an adaptation of a similar concept given by [Singh and Khim, 2022]:

Definition 7.1.1 (Admissible Binary Confusion Matrix Utility)**.** Let $\boldsymbol{c},\boldsymbol{c}'\in\mathcal{C}$. Then we say that $\boldsymbol{c}'$ is at least as good as $\boldsymbol{c}$, $\boldsymbol{c}'\succeq\boldsymbol{c}$, if there exist constants ϵ_1,ϵ_2 such that

$$\boldsymbol{c}' = [c_{\mathrm{tn}}+\epsilon_1, c_{\mathrm{fp}}-\epsilon_1, c_{\mathrm{fn}}-\epsilon_2, c_{\mathrm{tp}}+\epsilon_2], \tag{7.7}$$

i.e., if $\boldsymbol{c}'$ can be generated from $\boldsymbol{c}$ by turning some false positives to true negatives and false negatives to true positives. A function $\psi : \mathcal{C}\longrightarrow[0,1]$ is called a admissible binary confusion matrix utility if it respects that ordering, i.e., if for $\boldsymbol{c}'\succeq\boldsymbol{c}$ we have $\psi(\boldsymbol{c}')\geqslant\psi(\boldsymbol{c})$.

Similarly, in the multi-label case we cannot compare arbitrary confusion matrices, where one is better on some labels than on others, but we can recognize if

one is better on all labels:

Definition 7.1.2 (Admissible Multi-label Confusion Matrix Utility)**.** For a given number of labels $m \in \mathbb{N}$, and two confusion matrices $\mathbf{C}, \mathbf{C}' \in \mathcal{C}^m$, we say that $\mathbf{C}'$ is at least as good as $\mathbf{C}$, $\mathbf{C}' \succeq \mathbf{C}$, if for all labels $j \in [m]$ it holds that $\boldsymbol{c}'_j \geq \boldsymbol{c}_j$. A function $\Psi : \mathcal{C}^m \longrightarrow [0, 1]$ is called a confusion tensor measure if it respects this ordering, i.e., if for $\mathbf{C}' \succeq \mathbf{C}$ we have $\Psi(\mathbf{C}') \geqslant \Psi(\mathbf{C})$.

7.2 The optimal classifier in PU framework

Finding the form of the optimal classifier for general macro-averaged performance metrics is difficult. However, under mild assumptions on the data distribution, the optimal classifier exists and turns out to be the maximizer of some linear utility (5.2), whose coefficients, however, depend on its (unknown a priori) confusion matrix.

As we showed in Section 3.2 linear utilities, which are a special case of (5.1), have an appealingly simple form that boils down to simply sorting labels by an affine function of the marginals, and returning the top k elements, with ties broken arbitrarily. Because of the linearity of expectation, this is also true under PU framework with expected confusion matrix $\mathbf{C}(\boldsymbol{h}(\boldsymbol{x}))$. We consider this again because it will appear as a subproblem when finding optimal predictions for nonlinear macro-averages later in this section.

Theorem 7.2.1 (Regret for linear utilities under PU framework)**.** *The optimal classifier* $\boldsymbol{h}^\star \in \arg\max_{\boldsymbol{h} \in \mathcal{H}^{@k}} \Psi(\boldsymbol{h})$ *for* $\Psi(\boldsymbol{h}) = \sum_{j=1}^m \boldsymbol{g}_j \cdot \boldsymbol{c}_j$ *and any gain matrix* $\mathbf{G}$ *is given by:*

$$\boldsymbol{h}^\star(\boldsymbol{x}) \coloneqq \text{select-top-}k\left(\boldsymbol{a} \odot \boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{b}\right) , \tag{7.8}$$

where:

$$\boldsymbol{a} = \boldsymbol{g}_{:,\text{tn}} + \boldsymbol{g}_{:,\text{tp}} - \boldsymbol{g}_{:,\text{fp}} - \boldsymbol{g}_{:,\text{fn}} , \qquad \boldsymbol{b} = \boldsymbol{g}_{:,\text{fp}} - \boldsymbol{g}_{:,\text{tn}} . \tag{7.9}$$

Proof (sketch, full proof in Appendix A.4.2). The proof is very similar to the proof of Theorem 3.2.1 which presents the optimal classifier for general instance-wise weighted utilities@k. It follows the same algebraic manipulations. The objective can be written as $\Psi(\boldsymbol{h}) = \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}[\boldsymbol{x}]}\left[\sum_{j=1}^m (a_j \eta_j(\boldsymbol{x}) + b_j) h_j(\boldsymbol{x})\right] + r$, where r does not depend on the classifier. For each $\boldsymbol{x} \in \mathcal{X}$, the objective can thus be independently maximized by the choice of $\boldsymbol{h}(\boldsymbol{x}) \in \mathcal{H}^{@k}$ which maximizes $\sum_{j=1}^m (a_j \eta_j(\boldsymbol{x}) + b_j) h_j(\boldsymbol{x})$. □

For the class of randomized classifiers $\boldsymbol{h}^{\text{rnd}@k}$, this is simply achieved by sorting $a_j \eta_j(\boldsymbol{x}) + b_j$ in a descending order, and setting $\theta_j(\boldsymbol{x}) = 1$ for the top k coordinates, and $\theta_j(\boldsymbol{x}) = 0$ for the remaining coordinates (with ties broken arbitrarily), resulting in $\boldsymbol{\theta}(\boldsymbol{x}) = \text{select-top-}k\left(\boldsymbol{a} \odot \boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{b}\right)$. From now on, to the end of this chapter, we will use $\boldsymbol{h}$ to refer to classifiers belonging to the class of randomized classifiers $\boldsymbol{h}^{\text{rnd}@k}$. As we showed above, this includes linear deterministic classifiers as (7.8).

In Section 3.2, we already showed that precision@k, and Hamming score are linear utilities, however, we can also notice that macro-recall@k and macro balanced accuracy from Table 5.1 are also linear utilities under the PU framework, as the denominators in their definitions do not depend on the classifier. As in the case of the ETU framework, we can see that macro-recall@k is a linear utility with $\psi^j(\boldsymbol{c}_j) = \frac{c_{j,tp}}{\pi_j}$ and has $a_j = \frac{1}{\pi_j}, b_j = 0$, so that the optimal classifiers $\boldsymbol{h}^\star(\boldsymbol{x})$ returns top k labels sorted according to $\frac{\eta_j(\boldsymbol{x})}{\pi_j}$, where $\pi_j \coloneqq \mathbb{P}[y_i = 1]$. Also, the macro-balanced accuracy@k can be presented as $\psi^j(\boldsymbol{c}_j) = \frac{c_{tp}}{2\pi_j} + \frac{c_{tn}}{2(1-\pi_j)}$, giving $a_j = \frac{1}{2\pi_j} + \frac{1}{2(1-\pi_j)}, b_j = -\frac{1}{2(1-\pi_j)}$. The optimal classifier then sorts labels according to $\frac{\eta_j(\boldsymbol{x})}{\pi_j} - \frac{1-\eta_j(\boldsymbol{x})}{1-\pi_j}$.

Moving forward, we now switch to the general case, in which the base binary utility ψ^j are not necessarily decomposable over instances, and optimizing their macro averages with budgeted predictors is a challenging task. We make the following mild assumption on the data distribution.

Assumption 7.2.2. The label conditional marginal vector $\boldsymbol{\eta}(\boldsymbol{x}) = \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}]$ is absolutely continuous with respect to the Lebesgue measure on $[0,1]^m$ (i.e., has a density over $[0,1]^m$).

A similar assumption was commonly used in the past works [Koyejo et al., 2014, Narasimhan et al., 2015, Dembczyński et al., 2017].

Assumption 7.2.3. The performance metric Ψ is differentiable and fulfills for all labels $j \in [m]$:

$$\frac{\partial}{\partial \epsilon} \Psi(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_j + \epsilon[1,-1,-1,1], \ldots, \boldsymbol{c}_m)\Big|_{\epsilon=0} > 0\,. \tag{7.10}$$

Assumption 7.2.3 is essentially a strictly monotonic and differentiable version of Definition 7.1.2, and is satisfied by all macro-averaged metrics given in Table 5.1, except for coverage.

To state the main result concerning the form of the optimal classifier for general macro-averaged admissible binary confusion matrix measures, we need additionally to define $\mathcal{C}_{\mathbb{P}}^{m,@k} = \{\mathbf{C}(\boldsymbol{h}) : \boldsymbol{h} \in \mathcal{H}^{@k}\}$, the set of confusion matrices achievable by randomized k-budgeted classifiers on distribution $\mathbb{P}[\boldsymbol{x}, \boldsymbol{y}]$. Clearly, maximizing $\Psi(\boldsymbol{h})$ over $\boldsymbol{h} \in \mathcal{H}^{@k}$ is equivalent to maximizing $\Psi(\mathbf{C})$ over $\mathbf{C} \in \mathcal{C}_{\mathbb{P}}^{m,@k}$.

Theorem 7.2.4 (Regret for admissible multi-label utilities under the PU framework)**.** *Let the data distribution $\mathbb{P}[\boldsymbol{x}, \boldsymbol{y}]$ and metric Ψ satisfy Assumption 7.2.2 and Assumption 7.2.3, respectively. Then, there exists an optimal $\mathbf{C}^\star \in \mathcal{C}_{\mathbb{P}}^{m,@k}$, that is $\Psi(\mathbf{C}^\star) = \Psi^\star$. Moreover, any classifier $\boldsymbol{h}^\star$ maximizing the linear utility $\mathbf{G} \cdot \mathbf{C}(\boldsymbol{h})$ over $\boldsymbol{h} \in \mathcal{H}$ with $\mathbf{G} = [\boldsymbol{g}_1, \ldots, \mathbf{G}_m]$ given by $\boldsymbol{g}_j = \nabla_{\hat{\mathbf{c}}_j} \Psi(\hat{\boldsymbol{c}}^\star)$, also maximizes $\Psi(\boldsymbol{h})$ over $\boldsymbol{h} \in \mathcal{H}$.*

Proof (sketch, full proof in Appendix Appendix A.4.3. We first prove that $\mathcal{C}_{\mathbb{P}}^{m,@k}$ is a compact set by using certain properties of continuous linear operators in Hilbert space. Due to continuity of Ψ and the compactness of $\mathcal{C}_{\mathbb{P}}^{m,@k}$, there exists a maximizer $\mathbf{C}^\star = \arg\max_{\mathbf{C} \in \mathcal{C}_{\mathbb{P}}^{m,@k}} \Psi(\mathbf{C})$. By the first order optimality

and convexity of $\mathcal{C}_{\mathbb{P}}^{m,@k}$, $\nabla\Psi(\mathbf{C}^\star)\cdot\mathbf{C}^\star \geqslant \nabla\Psi(\mathbf{C}^\star)\cdot\mathbf{C}$ for all $\mathbf{C}\in\mathcal{C}_{\mathbb{P}}^{m,@k}$, so $\mathbf{C}^\star$ maximizes a linear utility $\mathbf{G}\cdot\mathbf{C}^\star$ with gain matrices given by $\mathbf{G}=\nabla\Psi(\mathbf{C}^\star)$. A careful analysis under Assumption 7.2.2 shows that $\mathbf{C}^\star$ uniquely maximizes $\mathbf{G}\cdot\mathbf{C}$ over $\mathbf{C}\in\mathcal{C}_{\mathbb{P}}^{m,@k}$. □

Theorem 7.2.4 reveals that Ψ-optimal classifier exists and can be found by maximizing a linear utility, that is, by predicting the top k labels sorted according to an affine function of the label marginals: $\boldsymbol{h}^\star(\boldsymbol{x}) = \text{select-top-}k(\boldsymbol{a}^\star \odot \boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{b}^\star)$ for vectors $\boldsymbol{a}^\star$ and $\boldsymbol{b}^\star$ defined for gain matrices $\mathbf{G}=\nabla\Psi(\mathbf{C}^\star)$ as in Theorem 7.2.1. Unfortunately, since $\mathbf{C}^\star$ is unknown in advance, coefficients $\boldsymbol{a}^\star, \boldsymbol{b}^\star$ are also unknown, and the optimal classifier is not directly available. However, knowing that $\boldsymbol{h}^\star$ optimizes a linear utility induced by the gradient of Ψ leads to a consistent algorithm described in the next section.

7.3 Consistent classifier via Frank-Wolfe

An algorithm we propose has to operate on a finite sample, so we go back to our initial empirical definitions of confusion matrix $\widehat{\mathbf{C}}$ (2.5) and empirical estimates of marginals $\widehat{\boldsymbol{\eta}}(\boldsymbol{x})$ given by some LPE trained on training dataset $\mathcal{D}_{\text{train}} = (\mathbf{X}, \mathbf{Y})$.

Following the work of Narasimhan et al. [2015], that focuses on the optimization of multi-class utilities, we use the Frank-Wolfe algorithm [Frank and Wolfe, 1956] to perform an implicit optimization over feasible confusion matrices $\mathcal{C}_{\mathbb{P}}^{m,@k}$, without having to explicitly construct $\mathcal{C}_{\mathbb{P}}^{m,@k}$. This is possible because Frank-Wolfe only requires us to be able to solve two sub-problems: First, given a classifier $\boldsymbol{h}$, we need to calculate its empirical confusion matrix, which is straightforward. Second, given a classifier and its corresponding confusion matrix, we need to solve a linearized version of the optimization problem, which is possible due to Theorem 7.2.1.

Consequently, our algorithm, which we present in Algorithm 7.1, proceeds as follows: In the beginning, we split the available training data into two subsets. One for estimating label probabilities $\widehat{\boldsymbol{\eta}}$, and one for tuning the actual classifier. After determining $\widehat{\boldsymbol{\eta}}$, we initialize $\boldsymbol{h}$ to be the standard top-k classifier, which will get iteratively refined as follows. For the confusion matrix of the current classifier, we can determine a linear objective based on its gradient. We then find a new linear classifier of the form as in (7.8) that maximizes this linearized utility. Because we can linearly interpolate stochastic classifiers, which will lead to linearly interpolated confusion matrices, we can obtain the direction of the descent along which we can optimize the step size,[1] and the confusion matrix for this classifier. We visualize this part of the algorithm in Figure 7.1. Based on this new confusion matrix, we can do the next linearized optimization step until we reach a stopping condition. We represent the randomized classifier as a set

[1]The classical version of the FW algorithm uses a fixed step-size schedule of $\frac{2}{t+1}$ instead of an inner optimization, but we find the latter to give better results empirically. However, for the convergence result, we assume fixed steps.

of deterministic classifiers $\boldsymbol{h}^t$ with corresponding weights α^t obtained across all iterations of the algorithm.

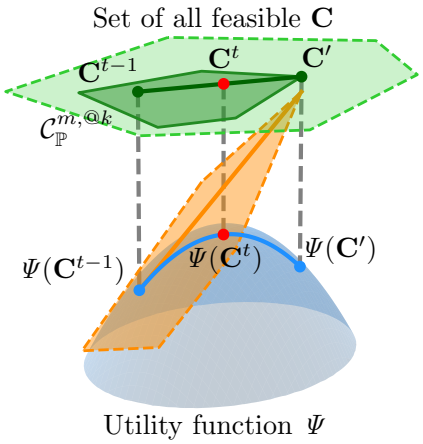


Figure 7.1: A visualization of a single step in the introduced Frank-Wolfe algorithm for optimization of multi-label utilities.

Algorithm 7.1 Frank-Wolfe algorithm for optimization of multi-label utilities

Require: a dataset $\mathcal{D}_{\text{train}}$, required budget k, utility function Ψ, stopping condition criterions ϵ and ϵ_α

Ensure: a randomized classifier $\boldsymbol{h}^{\text{rnd}@k}$

split dataset $\mathcal{D}_{\text{train}} = (\mathbf{X}, \mathbf{Y})$ into $\mathcal{D}_1 = (\mathbf{X}_1, \mathbf{Y}_1)$ and $\mathcal{D}_2 = (\mathbf{X}_2, \mathbf{Y}_2)$
learn label probability estimator $\widehat{\boldsymbol{\eta}} : \mathcal{X} \to \mathbb{R}^m$ on $\mathcal{D}_1$
initialize first classifier $\boldsymbol{h}^0 : \mathcal{X} \to \mathcal{Y}^{@k}$
calculate confusion matrix of the initial classifier $\widehat{\mathbf{C}}^0 \leftarrow \widehat{\mathbf{C}}(\mathbf{Y}_2, \boldsymbol{h}^0(\mathbf{X}_2))$
$\alpha^0 \leftarrow 1$
$t \leftarrow 0$
$u_{\text{old}} \leftarrow -\infty$
$u_{\text{new}} \leftarrow \Psi(\widehat{\mathbf{C}}^0)$
while $u_{\text{new}} > u_{\text{old}} + \epsilon \wedge \alpha^t > \epsilon_\alpha$ **do**
    $t \leftarrow t + 1$
    $\mathbf{G}^t \leftarrow \nabla_{\widehat{\mathbf{C}}} \Psi(\widehat{\mathbf{C}}^{t-1})$
    $\boldsymbol{a}^t \leftarrow \boldsymbol{g}^i_{:,\text{tp}} + \boldsymbol{g}^i_{:,\text{tn}} - \boldsymbol{g}^i_{:,\text{fp}} - \boldsymbol{g}^i_{:,\text{fn}}$
    $\boldsymbol{b}^t \leftarrow \boldsymbol{g}^i_{:,\text{fp}} - \boldsymbol{g}^i_{:,\text{tn}}$
    $\boldsymbol{h}^t(\boldsymbol{x}) \leftarrow \text{select-top-}k(\boldsymbol{a}^t \odot \boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{b}^t)$
    $\widehat{\mathbf{C}}' \leftarrow \widehat{\mathbf{C}}(\mathbf{Y}_2, \boldsymbol{h}^t(\mathbf{X}_2))$
    $\alpha^t \leftarrow \arg\max_{\alpha \in [0,1]} \Psi((1-\alpha)\widehat{\mathbf{C}}^{t-1} + \alpha \widehat{\mathbf{C}}')$
    $\widehat{\mathbf{C}}^t \leftarrow (1-\alpha^t)\widehat{\mathbf{C}}^{t-1} + \alpha^t \widehat{\mathbf{C}}'$
    for $t' \in \{0, \ldots, t-1\}$ **do**
        $\alpha^{t'} \leftarrow \alpha^{t'}(1-\alpha^t)$
    $u_{\text{old}} \leftarrow u_{\text{new}}$
    $u_{\text{new}} \leftarrow \Psi(\widehat{\mathbf{C}}^t)$
return $(\{\boldsymbol{h}^0, \ldots, \boldsymbol{h}^t\}, \{\alpha^0, \ldots, \alpha^t\})$

7.3.1 Computational complexity of the FW algorithm and resulting randomized classifier

The time and space complexity of the single step of Algorithm 7.1 are both $\mathcal{O}(|\mathcal{D}_2|m + m + t)$, assuming constant time for the gradient calculation of a single entry of the confusion matrix (line 9), and finding optimal step size α (line 14). Then predicting using $\boldsymbol{h}^t$ for $\mathbf{X}_2$ (line 13) have complexity of $\mathcal{O}(|\mathcal{D}_2|m)$, as top-$k$ selection can be performed in linear time using the introspective selection algorithm [Musser, 1997] and the rest of the operations require either $\mathcal{O}(m)$ (lines 9-11, 14-15) or $\mathcal{O}(t)$ (lines 16-17).

Predicting with randomized classifier $(\{\boldsymbol{h}^0, \ldots, \boldsymbol{h}^t\}, \{\alpha^0, \ldots, \alpha^t\})$ for a single instance $\boldsymbol{x}$ requires randomly selecting, according to probabilities $\{\alpha^0, \ldots, \alpha^t\}$, one of classifier $\boldsymbol{h}^s(\boldsymbol{x}) = \text{select-top-}k\,(\boldsymbol{a}^s \odot \boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{b}^s)$ resulting in worst case $\mathcal{O}(i+m)$ complexity. This complexity might be too high for extreme classification problems, where both n and m are very large. Because of that, we show a more efficient variant of the FW algorithm in Chapter 8.

7.3.2 Consistency of the FW algorithm

The introduced algorithm can consistently optimize the macro-averaged performance metric if it fulfills certain conditions:

Theorem 7.3.1 (Consistency of the Frank-Wolfe algorithm)**.** *Assume the utility function $\Psi : [0,1]^{m\times 4} \longrightarrow \mathbb{R}$ is concave over $\mathcal{C}_{\mathbb{P}}^{m,@k}$, L-Lipschitz, and β-smooth w.r.t the L_1-norm. Let $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ be a sample drawn i.i.d. from $\mathbb{P}$. Further, let $\widehat{\boldsymbol{\eta}}$ be a label probability estimator learned from $\mathcal{S}_1$, and $\boldsymbol{h}_{\mathcal{S}}^{\mathrm{FW}}$ be the classifier obtained after κn iterations. Then, for any $\delta \in (0,1]$, with probability of at least $1-\delta$ overdraws of $\mathcal{S}$,*

$$\begin{aligned}\mathrm{Reg}(\boldsymbol{h}_{\mathcal{S}}^{\mathrm{FW}}) \leqslant\ & \mathcal{O}\left(\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_1\right) \\ & + \tilde{\mathcal{O}}\left(m^2\sqrt{\frac{m\cdot\log m\cdot\log n - \log\delta}{n}}\right) + \frac{8\beta m}{\kappa n + 2}\,. \end{aligned} \tag{7.11}$$

Proof (sketch, full proof in Appendix A.4.4). The proof broadly follows a similar result of Narasimhan et al. [2015]. First, we show that linear metrics can be estimated consistently with a regret growing with the L_1 error of the LPE. Then, we prove a uniform convergence result for estimating the multi-label confusion matrix. As a prerequisite, we derive the VC-dimension of the class of classifiers based on top-k scoring, i.e., those classifiers that minimize some linear confusion matrix metric as shown in Theorem 7.2.1. We derive an upper bound of $6m\log(em)$ as shown in the lemma below.

Lemma 7.3.2 (VC dimension for linear top-k classifiers)**.** *For $\boldsymbol{\eta} : \mathcal{X} \longrightarrow [0,1]^m$, define the hypothesis class*

$$\mathcal{H}_{\boldsymbol{\eta}}^{j} := \bigcup_{\boldsymbol{a},\boldsymbol{b}\in\mathbb{R}^m} \{h : \mathcal{X} \longrightarrow \{0,1\} \,:\, h(\boldsymbol{x}) = \mathbb{1}\left[j \in \arg\text{top-}k\ (\boldsymbol{a}\odot\boldsymbol{\eta} + \boldsymbol{b})\right]\}\,. \tag{7.12}$$

The VC-complexity of this class is

$$VC(\mathcal{H}^j_{\boldsymbol{\eta}}) \leqslant 6m \log(em)\,. \tag{7.13}$$

To prove this lemma, we first notice that for any given $\boldsymbol{a}, \boldsymbol{b}$, the hypothesis predicts one, $h^j(\boldsymbol{x}) = 1$, iff exists a set of $m-k$ indices $\mathcal{I} \subset [m]$ with $|\mathcal{I}| = m-k$, $j \notin \mathcal{I}$, such that for all $i \in \mathcal{I}$ the score $a_i\eta_i + b_i \leqslant a_j\eta_j + b_j$ is not greater than the score of label j.

Then we show that this computation can be realized as a two-layer network with $2(m-1)$ edges and $m-1$ computation nodes. If we allow the output node to be more general – a generic linear threshold function, the VC-dimension of this extended function class $\mathcal{H}'$ can only grow. For this extended class, we can apply [Baum and Haussler, 1988, Corollary 3], which gives an upper bound for the VC-dimension.

With this upper bound of the VC dimension, we can apply standard arguments for the convergence of the Frank-Wolfe algorithm. □

Frank-Wolfe algorithm for metrics without a budget constraint

The introduced Frank-Wolfe algorithm can also be used to optimize label-wise metrics without the budget k constraint. Since all utilities used in practice are non-decreasing with true positives and true negatives ($g_{j,\text{tp}}, g_{j,\text{tn}} \geqslant 0$), and non-increasing with false negatives and false positives ($g_{j,\text{fn}}, g_{j,\text{fp}} \leqslant 0$), we get $a_j \geqslant 0$, and thus maximizing $\sum_{j=1}^{m}(a_j\eta_j(\boldsymbol{x}) - b_j)h_j(\boldsymbol{x})$ boils down to choosing $\widehat{y}_j = 1$ whenever the conditional probability $\eta_j(\boldsymbol{x})$ exceeds a threshold $\frac{-b_j}{a_j}$. Using the above observation, one can prove that thresholding $\boldsymbol{\eta}(\boldsymbol{x})$ is the optimal classification rule for maximizing Ψ under Population Utility framework [Koyejo et al., 2014, Narasimhan et al., 2014], under mild assumptions on the data distribution and utility function Ψ.

7.4 Summary of the chapter

In this chapter, we have analyzed the optimization problem of metrics budgeted at k, being functions of a multi-label confusion matrix, which also include label-wise utilities, in the population utility (PU) framework. We have first shown that under the PU framework, macro-recall@k and macro-balanced accuracy@k are linear utilities. Then, we have demonstrated that under the conditions of the conditional marginal distribution $\boldsymbol{\eta}(\boldsymbol{x})$ to be absolutely continuous and of utility $\Psi(\widehat{\mathbf{C}})$ to be strictly monotonic and differentiable, which is satisfied by most utilities of interest, the optimal classifier is a linear function of a confusion matrix. Next, we have introduced a consistent Frank-Wolfe (FW) algorithm, which is capable of finding an optimal randomized classifier by transforming the problem of optimizing over classifiers to a problem of optimizing over the set of feasible confusion matrices and

by using the fact that the optimal classifier optimizes (unknown) linear confusion matrix. In Chapter 9, we present the empirical results of the FW, combined with the state-of-the-art label probability estimator for extreme multi-label classification, demonstrating that it is an effective and efficient method for the optimization of label-wise utilities in the PU framework.

Beyond label-wise utilities in PU and ETU frameworks

Notice that the introduced Frank-Wolfe algorithm can also be applied to utilities that do not decompose over labels, it only requires for the utility $\Psi(\mathbf{C})$ to be differentiable in $\mathbf{C}$.

While in this work we keep our analysis constrained to label-wise utilities (5.1), it is worth noticing that a similar linearization of the objective can be applied to the Block Coordinate Algorithm introduced in Section 6.5. Directly calculating the expected gain of predicting a single label (6.23) can be done in linear time only for label-wise utilities. For other types of metrics, one can calculate the gradient of the current ETU approximation of the confusion matrix without the currently considered instances and select the prediction that maximizes the linearized utility of the function:

$$\widehat{\boldsymbol{y}}_s = \arg\max_{\widehat{\boldsymbol{y}}} \nabla_{\tilde{\mathbf{C}}}\Psi\left(\tilde{\mathbf{C}}\left(\widehat{\mathbf{Y}}_{\{1,\ldots,n\}\setminus s}\right)\right) \cdot \mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\tilde{\mathbf{C}}\left(\widehat{\boldsymbol{y}}_s\right)\right]. \tag{7.14}$$

By abbreviating $\nabla_{\tilde{\mathbf{C}}}\Psi\left(\tilde{\mathbf{C}}\left(\widehat{\mathbf{Y}}_{\{1,\ldots,n\}\setminus s}\right)\right)$ as $\mathbf{G}$, we can write down (7.14) as:

$$\widehat{\boldsymbol{y}}_s = \arg\max_{\widehat{\boldsymbol{y}}} \sum_{j=1}^{m} (a_j\eta_j(\boldsymbol{x}) + b_j)\widehat{y}_j\,, \tag{7.15}$$

with $a_j = g_{j,\mathrm{tp}} + g_{j,\mathrm{tn}} - g_{j,\mathrm{fp}} - g_{j,\mathrm{fn}}$ and $b_j = g_{j,\mathrm{fp}} - g_{j,\mathrm{tn}}$. Once again, maximizing (7.15) corresponds to setting $\widehat{y}_j = 1$ whenever $\eta_j(\boldsymbol{x}) \geqslant \frac{-b_j}{a_j}$, or in case of prediction budgeted at k, selecting of k labels with highest values of $a_j\eta_j(\boldsymbol{x}) + b_j$.

The same generalization can be applied to the Greedy algorithm (Section 6.8), resulting in a general online algorithm for complex utilities. Such an algorithm has been studied in detail in [Kotłowski et al., 2024].

8

Efficient algorithms for at k prediction

In this chapter, we introduce methods that allow for efficient inference using algorithms presented in Chapters 3, 6 and 7 in the setting of extreme multi-label classification. We start the chapter with a discussion on how we can leverage the sparsity of the label space to reduce the time and space complexity of the inference algorithms, generalizing the analysis presented in Schultheis et al. [2023]. Then, we introduce the idea of probabilistic label trees (PLTs) [Jasinska et al., 2016] that can be used as an output layer for an arbitrary neural network architecture and allow efficient training and prediction [Wydmuch et al., 2018, 2021].

8.1 Reducing the complexity of the algorithms via compressed sparse representations

The algorithms introduced so far in Chapters 3, 6 and 7 need to obtain estimates $\widehat{\boldsymbol{\eta}}(\boldsymbol{x})$ of all conditional marginal probabilities. After obtaining them, the time and space complexity of the inference is of order $\mathcal{O}(m)$ for each instance $\boldsymbol{x}$ in the case of all presented algorithms, that is, for the optimal prediction for instance-wise weighted utilities and (n)DCG@k (Chapter 3), for the single update in the BCA algorithm (Chapter 6), and for the prediction with the randomized classifier obtained via the Frank-Wolfe algorithm (Chapter 7). As mentioned in Chapter 1, this is problematic in the setting of XMLC, where many methods aim to predict precise probabilities only for a few top labels in time sublinear to m. This characteristic of XMLC algorithms can be combined with the introduced algorithms to efficiently obtain an approximate solution.

Instead of predicting all $\widehat{\boldsymbol{\eta}}(\boldsymbol{x})$, we can focus only on top-k' labels with the highest $\widehat{\eta}_j(\boldsymbol{x})$, where $k \ll k' \ll m$. For all other labels, we then assume $\widehat{\eta}_j(\boldsymbol{x}) = 0$. Under the reasonable assumption on utility function Ψ being non-decreasing with true positives and true negatives, and non-increasing with false negatives and false

positives, we leverage the sparsity of $\widehat{\boldsymbol{\eta}}(\boldsymbol{x})$ by considering labels with non-zero $\widehat{\eta}_j(\boldsymbol{x})$ only to reduce the time and space complexity of the inference algorithms.

Let us recall that for the algorithms discussed in Chapters 3, 6 and 7, we introduce regret bounds that depend on either L_1 (Theorems 3.2.2, 3.4.1, 6.7.1 and 7.3.1) or L_2 (Theorem 6.6.1) error of the predicted conditional marginal probabilities of labels. As in real-world datasets, the number of relevant labels $\|\boldsymbol{y}\|_1$ is much lower than m, and most $\eta_j(\boldsymbol{x})$ are close to 0. With reasonably selected k', according to these bounds, reducing values that are close to 0 to actual 0 may only slightly increase regret, as $\eta_j(\boldsymbol{x})$ estimation error change is minimal. At the same time, predicting only top-k' labels and their conditional probabilities is, by design, much more efficient in many XMLC methods. Alternatively, instead of predicting fixed top-k' for each instance, some XMLC methods allow for the prediction of all labels whose probability estimates exceed a given threshold. While this potentially allows for more precise control of regret, for the simplicity of the analysis of the algorithms' complexity, we keep the assumption that k' is the same for each instance.

To take advantage of the sparsity of labels, one can use a compressed sparse (CS) representation of vectors. The sparse representation of vector $\boldsymbol{a}$ of size[1] $|\boldsymbol{a}|$ consists of a list of tuples with the index and value of non-zero elements $\text{cs}(\boldsymbol{a}) \coloneqq [(i, a_i)_{i=1}^{|\boldsymbol{a}|} : a_i \neq 0]$, with $|\text{cs}(\boldsymbol{a})|$ denoting the number of tuples in the list (non-zero elements of a vector $\boldsymbol{a}$). We also use $\cup$ symbol to denote the concatenation of lists. Let us notice that under this representation, the basic vector operations have the following complexities:

- an operation of addition of a dense vector and a sparse vector $(\boldsymbol{a} + \text{cs}(\boldsymbol{b}))$ has a complexity of $\mathcal{O}(|\boldsymbol{a}|)$ or $\mathcal{O}(|\text{cs}(\boldsymbol{b})|)$ for in-place addition,
- an operation of addition of a sparse vector and a sparse vector $(\text{cs}(\boldsymbol{a}) + \text{cs}(\boldsymbol{b}))$ has a complexity of $\mathcal{O}(|\text{cs}(\boldsymbol{a})| + |\text{cs}(\boldsymbol{b})|$,
- a dot product and Hadamard (element-wise) products of a dense vector and a sparse vector, which return another sparse vector $(\boldsymbol{a} \cdot \text{cs}(\boldsymbol{b}), \boldsymbol{a} \odot \text{cs}(\boldsymbol{b}))$ have complexities of $\mathcal{O}(|\text{cs}(\boldsymbol{b})|$,
- a dot and Hadamard products of a sparse vector with a sparse vector $(\text{cs}(\boldsymbol{a}) \cdot \text{cs}(\boldsymbol{b}), \text{cs}(\boldsymbol{a}) \odot \text{cs}(\boldsymbol{b}))$ have complexity $\mathcal{O}(|\text{cs}(\boldsymbol{a})| + |\text{cs}(\boldsymbol{b})|$, if both $\text{cs}(\boldsymbol{a}$, and $\text{cs}(\boldsymbol{b})$ are sorted according to their indices. Otherwise, the complexity turns into $\mathcal{O}(|\text{cs}(\boldsymbol{a})| \log |\text{cs}(\boldsymbol{a})| + |\text{cs}(\boldsymbol{b})| \log |\text{cs}(\boldsymbol{b})|)$.

In the setting of XMLC, the label space is so large that $\mathbf{Y}$ and $\widehat{\mathbf{Y}}$, in practice, are almost always represented as matrices in the compressed sparse row (CSR) format, in which each row of a matrix is stored in the compressed sparse vector representation. Now we additionally assume that $\widehat{\boldsymbol{\eta}}(\boldsymbol{x})$ is also in the compressed sparse format with exactly k' non-zero elements, i.e., $|\text{cs}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}))| = k'$. As we predict k labels for each instance, the resulting matrix $\widehat{\mathbf{Y}}$ has each row of exactly k non-zero elements (ones), i.e., $\forall_{i \in [n]} |\text{cs}(\widehat{\boldsymbol{y}}_i)| = k$.

[1] Please note that we use set notation for the number of elements in a vector (i.e., the size of a vector). The same notation is applied to the compressed sparse representation of vectors, which is a list.

Under the additional assumption that we never predict labels with $\eta_j(\boldsymbol{x}) = 0$, we can reduce the complexity of the select-top-$k(\boldsymbol{a} \odot \boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{b})$ function by using compressed sparse representation to $\mathcal{O}(k')$ and $\mathcal{O}(k' + k)$ for time and space, respectively. This is because $\boldsymbol{a} \odot \boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{b}$ requires $\mathcal{O}(k')$ operations (with b being added only to non-zero entries) and selection of top k can be performed using the introspective selection algorithm [Musser, 1997] which has $\mathcal{O}(k')$ time and $\mathcal{O}(k' + k)$ space complexity. We present this procedure in Algorithm 8.1.

Algorithm 8.1 Sparse select-top-$k(\boldsymbol{a} \cdot \boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{b})$

Require: labels marginal probability estimates for an instance $\boldsymbol{\eta}(\boldsymbol{x})$, required budget k, vectors $\boldsymbol{a}$ and $\boldsymbol{b}$
Ensure: a sparse prediction vector $\operatorname{cs}(\widehat{\boldsymbol{y}})$ with k non-zero entries
1: $\operatorname{cs}(\boldsymbol{g}) \leftarrow \varnothing$
2: **for** $(j, \eta_j) \in \operatorname{cs}(\boldsymbol{\eta}(\boldsymbol{x}))$ **do**
3: $\quad \operatorname{cs}(\boldsymbol{g}) \leftarrow \operatorname{cs}(\boldsymbol{g}) \cup (j, a_j \eta_j + b_j)$
4: $\operatorname{cs}(\widehat{\boldsymbol{y}}) \leftarrow$ select-top-$k(\operatorname{cs}(\boldsymbol{g}))$
5: **return** $\operatorname{cs}(\widehat{\boldsymbol{y}})$

Additionally, we are interested in the complexity of calculating the entries of a confusion matrix, used in the BCA as well as in the Frank-Wolfe algorithm, under the sparse representation of labels $\operatorname{cs}(\boldsymbol{y})$, predictions $\operatorname{cs}(\widehat{\boldsymbol{y}})$, and estimated conditional marginal probabilities $\operatorname{cs}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}))$. Let us start with BCA, which first needs to calculate the initial confusion matrix in the alternative parametrization, and then update the confusion matrix for every changed prediction:

- Calculation of the true positives ratio for all labels $\tilde{\boldsymbol{c}}_{:,\mathrm{tp}} = \frac{1}{n} \sum_{i=1}^{n} \operatorname{cs}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}_i)) \odot \operatorname{cs}(\widehat{\boldsymbol{y}}_i)$ has time and space complexity $\mathcal{O}\left(n(k' + k)\right)$ for a set of n instances, and updating it (subtracting and adding for change in $\operatorname{cs}(\widehat{\boldsymbol{y}}_i)$) for a single instance $\boldsymbol{x}_i$ is $\mathcal{O}\left(k' + k\right)$.
- Calculation of predicted positives $\widehat{\boldsymbol{c}}_{:,\mathrm{pp}} = \frac{1}{n} \sum_{i=1}^{n} \operatorname{cs}(\widehat{\boldsymbol{y}}_i)$ has simply complexity of $\mathcal{O}(nk)$ and $\mathcal{O}(k)$ for an update.
- Finally, calculation of conditional positives $\tilde{\boldsymbol{c}}_{:,\mathrm{cp}} = \frac{1}{n} \sum_{i=1}^{n} \operatorname{cs}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}_i))$ has complexity $\mathcal{O}(nk')$; this value is computed only once in the BCA algorithm.

Under a similar assumption as in the case of sparse selection of top-k labels (Algorithm 8.1), that an algorithm does not predict labels with marginal probability equal to 0, the time and space complexity of a single iteration of BCA algorithm reduces from $\mathcal{O}\left(nm + m\right)$ to $\mathcal{O}\left(n(k' + k) + k' + k\right)$.

Now let us move to the Frank-Wolfe algorithm, which in every step needs to calculate the confusion matrix for the validation set $\mathcal{D}_2$ and predictions of a new weighted classifier, $\mathbf{C}(\mathbf{Y}_2, \boldsymbol{h}^i(\mathbf{X}_2))$:

- Calculation of the true positives ratio for all labels $\widehat{\boldsymbol{c}}_{:,\mathrm{tp}} = \frac{1}{n} \sum_{i=1}^{n} \operatorname{cs}(\boldsymbol{y}_i) \odot \operatorname{cs}(\widehat{\boldsymbol{y}}_i)$ has complexity $\mathcal{O}\left(|\mathcal{D}_2|(\mathbb{E}_{\boldsymbol{y} \sim \mathbb{P}[\boldsymbol{y} \mid \boldsymbol{x}]}[|\operatorname{cs}(\boldsymbol{y})|] + k)\right)$ for set of $|\mathcal{D}_2|$ instances.
- Calculation of false positives $\widehat{\boldsymbol{c}}_{:,\mathrm{fp}} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{1} - \operatorname{cs}(\boldsymbol{y}_i)) \odot \operatorname{cs}(\widehat{\boldsymbol{y}}_i)$ has the same complexity as above, because of the distributive property of the Hadamard

product, i.e., $(\mathbf{1} - \operatorname{cs}(\boldsymbol{y}_i)) \odot \operatorname{cs}(\widehat{\boldsymbol{y}}_i) = \mathbf{1} \odot \operatorname{cs}(\widehat{\boldsymbol{y}}_i) - \operatorname{cs}(\boldsymbol{y}_i) \odot \operatorname{cs}(\widehat{\boldsymbol{y}}_i)$.

- Calculation of false negatives $\widehat{\boldsymbol{c}}_{:,\mathrm{fn}} = \frac{1}{n}\sum_{i=1}^{n} \operatorname{cs}(\boldsymbol{y}_i) \odot (\mathbf{1} - \operatorname{cs}(\widehat{\boldsymbol{y}}_i))$ has also the same complexity by the same argument as above.
- Calculation of true negatives $\widehat{\boldsymbol{c}}_{:,\mathrm{fn}} = \frac{1}{n}\sum_{i=1}^{n} (\mathbf{1} - \operatorname{cs}(\boldsymbol{y}_i) \odot (\mathbf{1} - \operatorname{cs}(\widehat{\boldsymbol{y}}_i))$ has, unfortunately, the complexity of $\mathcal{O}(m)$ as the distributive property of the Hadamard product does not help to reduce it in this case. However, as true negatives are redundant in the presence of all the other entries of the confusion matrix, they can be easily calculated by subtracting true positive, false negative, and false positive rates from a vector of ones, in case they are needed.

Combining this with the efficient selection of top-k labels by a classifier, the time and space complexity of a single step of Frank-Wolfe (Algorithm 7.1) reduces from $\mathcal{O}\left(|\mathcal{D}_2|m + m + i\right)$ to $\mathcal{O}\left(|\mathcal{D}_2|(\mathbb{E}_{\boldsymbol{y} \sim \mathbb{P}[\boldsymbol{y} \mid \boldsymbol{x}]}[|\operatorname{cs}(\boldsymbol{y})|] + k' + k) + m + t\right)$.

In empirical experiments in Chapter 9, we demonstrate that indeed selecting from top-k', with $k' = 100$ has minimal impact on predictive performance, while it reduces time and memory requirements by orders of magnitude.

8.2 Efficient training and inference with a large number of labels via probabilistic label trees

In the previous section, we discussed a general method of reducing the time and space complexity of the algorithms considered in this work, at the risk of suffering a small regret. In this section, we discuss an algorithm for training a probability estimator $\boldsymbol{\eta}(\boldsymbol{x})$ that reduces both the complexity of training and allows for efficient exact prediction for select-top-$k(\boldsymbol{a} \odot \boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{b})$, possibly alleviating the issue of suffering regret when considering only top-k' labels, while simultaneously being faster when aiming to predict $k \ll k'$ labels.

To achieve this, we consider a class of methods referred to as probabilistic label trees (PLTs) [Jasinska et al., 2016], which is a popular approach in XMLC with many variants and improvements [Prabhu et al., 2018b, Wydmuch et al., 2018, Khandagale et al., 2020, You et al., 2019a, Chang et al., 2020, Jasinska-Kobus et al., 2020, Ye et al., 2020, Jiang et al., 2021, Zhang et al., 2021, Yu et al., 2022] due to its high predictive and computational performance, while simultaneously being a pure multi-label approach that does not require any additional information about labels.

PLTs efficiently solve the problem of estimating marginal label probabilities in multi-label classification. To this end, they reduce the original problem to a set of binary problems organized in the form of a tree, to which we refer as a label tree. In the case of standard multi-label classification, the labels are assigned to m leaf nodes, with each label corresponding to one and only one node. PLTs can also naturally support hierarchical multi-label classification. In this case, the labels are also assigned to non-leaf nodes.

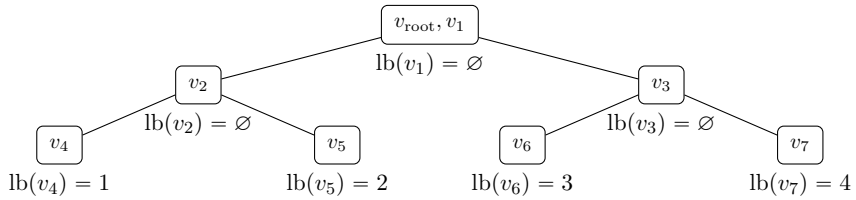


Figure 8.1: A label tree with 4 labels assigned to leave nodes. For example, in this tree: $\text{Path}(v_4) = \{v_1, v_2, v_4\}$, $\text{Ch}(v_1) = \{v_2, v_3\}$, $\text{Labels}(v_2) = \{1, 2\}$, $\text{pa}(v_4) = v_2$, $\text{v}(1) = v_4$.

We denote a tree by $\mathcal{T}$ that consists of a set of its nodes (vertices), denoted as $\mathcal{V}$, which are connected by edges. The root node is denoted by v_{root}, a parent node of v by $\text{pa}(v)$, and the set of child nodes by $\text{Ch}(v)$. A node can have a single label assigned to it, the assigned label to node v is denoted as $\text{lb}(v)$, and a node corresponding to label j as $\text{v}(j)$. Each label $j \in [m]$ is assigned to exactly one unique node in $\mathcal{V}$. The set of labels assigned to the nodes in (sub)tree rooted in node v is denoted by $\text{Labels}(v)$, and the path from node v to the root by $\text{Path}(v)$. An example of a label tree and all the above operators can be found in Figure 8.1.

PLT uses tree $\mathcal{T}$ to factorize the marginal conditional probabilities of labels $\eta_j(x)$ by using the chain rule. Let us define an event that $\text{Labels}(v)$ contains at least one relevant label from $\boldsymbol{y}$:

$$z_v = \mathbb{1}\left[\sum_{j \in \text{Labels}(v)} y_j > 0\right]. \tag{8.1}$$

Now for every node $v \in \mathcal{V}$, the marginal conditional probability of containing at least one relevant label $\eta_v^{\mathcal{T}}$ is given by:

$$\eta_v^{\mathcal{T}}(\boldsymbol{x}) = \mathbb{P}[z_v = 1 | \boldsymbol{x}] = \prod_{v' \in \text{Path}(v)} \eta^{\mathcal{T}}(\boldsymbol{x}, v'), \tag{8.2}$$

where $\eta^{\mathcal{T}}(\boldsymbol{x}, v) = \mathbb{P}[z_v = 1 | z_{\text{pa}(v)} = 1, \boldsymbol{x}]$ for non-root nodes, and $\eta^{\mathcal{T}}(\boldsymbol{x}, v) = \mathbb{P}(z_v = 1 \,|\, \boldsymbol{x})$ for the root. Notice that (8.2) can also be stated as a recursion:

$$\eta_v^{\mathcal{T}}(\boldsymbol{x}) = \eta_{\text{pa}(v)}^{\mathcal{T}}(\boldsymbol{x}) \eta^{\mathcal{T}}(\boldsymbol{x}, v). \tag{8.3}$$

For nodes with assigned labels, we get the marginal conditional probabilities of labels:

$$\eta_j(\boldsymbol{x}) = \eta_{\text{v}(j)}^{\mathcal{T}}(\boldsymbol{x}). \tag{8.4}$$

The idea of decomposing the conditional probability of a label was introduced in the context of multi-class classification, and is known in the literature as the nested dichotomies [Fox, 1997], hierarchical softmax [Morin and Bengio, 2005], conditional probability estimation trees [Beygelzimer et al., 2009a], or probabilistic classifier trees [Dembczyński et al., 2016].

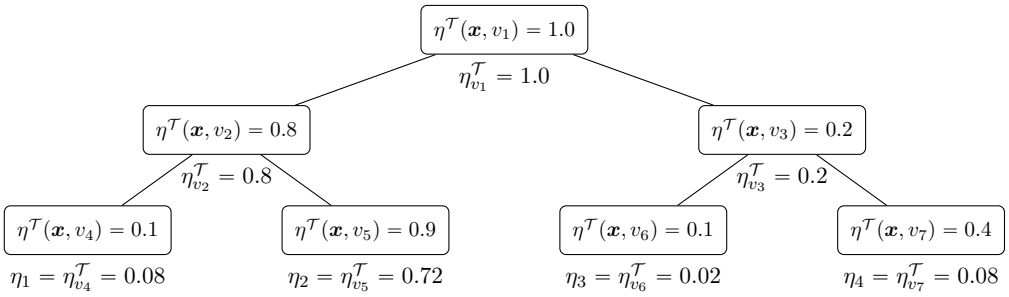


Figure 8.2: An example of decomposition of marginal conditional probabilities by probabilistic label tree (PLT).

8.2.1 Training PLTs

To obtain a PLT, it suffices for a given $\mathcal{T}$ to train probability estimators of $\eta^{\mathcal{T}}(\boldsymbol{x}, v)$ for all $v \in \mathcal{V}$. Because of the factorization (8.2), a probability estimator in each node might be trained using only a subset of the entire training set $\mathcal{D}_{\text{train}} := [(\boldsymbol{x}_i, \boldsymbol{y}_i)]_{i=1}^{n_{\text{train}}}$.

The training algorithm relies on a proper assignment of training samples $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ to nodes to correctly learn their probability estimators, as given in (8.2). For each sample with labels $\boldsymbol{y}$, the algorithm identifies a set of positive $\mathcal{P}(\boldsymbol{y})$ and negative nodes $\mathcal{N}(\boldsymbol{y})$, i.e., the nodes for which a training example is treated, respectively as positive (i.e., $(\boldsymbol{x}, z_v = 1)$) or negative (i.e., $(\boldsymbol{x}, z_v = 0)$). The procedure is given in Algorithm 8.2. It initializes the positive nodes to the empty set and the negative nodes to the root node (to deal with $\boldsymbol{y}$ being all zeros). Next, it traverses the tree from the leaves corresponding to the labels of the training example to the root, adding the visited nodes to the set of positive nodes. It also removes each visited node from the set of negative nodes if it has been added to this set before. All children of the visited node, which are not in the set of positive nodes, are then added to the set of negative nodes. If the parent node of the visited node has already been added to positive nodes, the traversal on this path stops.

Algorithm 8.2 Assign positive and negative nodes to sample

Require: any tree structure $\mathcal{T}$, a vector of labels $\boldsymbol{y}$
Ensure: a set of positive and negative nodes that need to be updated for $\boldsymbol{y}$ in $\mathcal{T}$

$\mathcal{P}(\boldsymbol{y}) \leftarrow \varnothing$
$\mathcal{N}(\boldsymbol{y}) \leftarrow \{v_{\text{root}}\}$
for $j \in \{j : y_j = 1\}$ **do**
  $vv \leftarrow \text{v}(j)$
  while $v \neq \varnothing$ **and** $v \notin \mathcal{P}(\boldsymbol{y})$ **do**
    $\mathcal{P}(\boldsymbol{y}) \leftarrow \mathcal{P}(\boldsymbol{y}) \cup \{v\}$
    $\mathcal{N}(\boldsymbol{y}) \leftarrow \mathcal{N}(\boldsymbol{y}) \backslash \{v\}$
    for $v' \in \text{Ch}(v)$ **do**
      if $v' \notin \mathcal{P}(\boldsymbol{y})$ **then** $\mathcal{N}(\boldsymbol{y}) \leftarrow \mathcal{N}(\boldsymbol{y}) \cup \{v'\}$
    $v \leftarrow \text{pa}(v)$
return $\mathcal{P}(\boldsymbol{y}), \mathcal{N}(\boldsymbol{y})$

Notice that if we assume a complete tree with arity r (tree with all levels

fully filled, with each node having exactly r children, except on the last level), the number of total updated nodes for each sample $(\boldsymbol{x}, \boldsymbol{y})$ is upper-bounded by $\|\boldsymbol{y}\|_1 r \log_r m$. This number can possibly be lower, depending on how many nodes are shared between paths from positive label nodes to the tree roots. Since in XMLC $\|y\|_1 \ll m$, the computational complexity of training PLT is much lower than training flat classifiers that do m updates for every instance.

Of course, in general, the tree does not need to be complete or have the same arity for every node. The tree can be a predefined hierarchy of labels, however in most cases, where the original structure of labels is flat, the artificial tree is created on top of labels. While the original PLT was using a random complete tree [Jasinska et al., 2016], it has been shown that the tree structure that places the similar (i.e., co-occurring) labels close in the tree helps to learn good node probability estimators $\widehat{\eta}(\boldsymbol{x}, v)$ [Prabhu et al., 2018b]. Many different tree-building techniques have been proposed, with the most popular involving hierarchical clustering of labels into smaller subsets based on different representations of labels [Prabhu et al., 2018b, Khandagale et al., 2020, Yu et al., 2022]. Shallow trees with a large arity of nodes have become popular, as they are much more efficient to use on GPUs [Jiang et al., 2021]. Busa-Fekete et al. [2019] considered the problem of building the tree that will be the most efficient during the training or prediction, showing the problem to be NP-complete in general. Jasinska-Kobus et al. [2021] have proposed a method for building a tree in an online fashion without the need to know the set of labels upfront.

8.2.2 PLT as an output layer of a neural network

Notice that all node probability estimators can be, in fact, trained independently. Consequently, the original implementation, as well as some of the following variants of PLTs, used independent methods for training probability estimators, e.g., logistic regression. However, since the number of nodes is greater than the number of labels $|\mathcal{V}| > m$, this limits the usage of larger independent models, e.g., deep neural networks as node estimators, due to their space requirements. Moreover, the usage of simple methods may limit the training of accurate node estimators.

To enable efficient learning of neural networks, inspired by the application of hierarchical softmax [Goodman, 2001] as a loss layer for learning neural networks for multi-class classification [Morin and Bengio, 2005, Joulin et al., 2017], Wydmuch et al. [2018] have proposed using PLT as an output layer and loss function to optimize the parameters of any underlying neural network. In the PLT output layer, all node probability estimators are represented as a single matrix of parameters $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}| \times |\boldsymbol{e}(\boldsymbol{x})|}$ and bias vector $\boldsymbol{b} \in \mathbb{R}^{|\mathcal{V}|}$, where $\boldsymbol{e}(\boldsymbol{x})$ is a latent representation vector of an instance $\boldsymbol{x}$ used as input to the PLT output layer. The node probability estimates are then simply calculated as:

$$\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v) = \sigma(\boldsymbol{w}_v \cdot \boldsymbol{e}(\boldsymbol{x}) + b_v)\,, \tag{8.5}$$

where $\sigma(\cdot)$ is a standard sigmoid function $\sigma(x) = 1/(1 - e^{-1})$. The PLT loss is

then calculated for each instance $\boldsymbol{x}$ as:

$$\ell_{\mathrm{PLT}}(\boldsymbol{y}, \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, \cdot)) \coloneqq \sum_{v \in \mathcal{P}(\boldsymbol{y})} \log \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v) + \sum_{v \in \mathcal{N}(\boldsymbol{y})} \log(1 - \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v)) \,, \tag{8.6}$$

with $\mathcal{P}(\boldsymbol{y})$ and $\mathcal{N}(\boldsymbol{y})$ obtained using Algorithm 8.2. The parameters $\mathbf{W}$, as well as the rest of the parameters of the underlying neural network, can then be updated using a gradient $\nabla \ell_{\mathrm{PLT}}$ in the standard backpropagation [Kelley, 1960]. Notice that only positive and negative nodes contribute to the loss value, and for every instance $\boldsymbol{x}$, only weights $\boldsymbol{w}_v$, for $v \in \mathcal{P}(\boldsymbol{y}) \cup \mathcal{N}(\boldsymbol{y})$, need to be updated, reducing the complexity of calculating the loss and of the last layer of a neural network in comparison to the standard flat output and loss function, such as sigmoid output and binary cross-entropy loss.

8.2.3 Efficient prediction with PLT

A true structure of PLT allows for applying a tree search algorithms for efficiently finding a set of labels exceeding the given threshold τ or a set of labels with top-k probabilities [Jasinska et al., 2016].

In the first case, we can apply any standard tree traverse algorithm, e.g., in a depth-first or breadth-first manner. The algorithm starts traversal with the root node v_{root} and adds children nodes to a simple first-in-first-out queue $\mathcal{Q}$ of nodes to visit if their conditional marginal probability $\widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x})$ exceeds threshold τ. Once the queue is empty, the algorithms return all visited nodes with assigned labels, resulting in a positive prediction of all labels with $\widehat{\eta}_j(\boldsymbol{x}) > \tau$. We outline this procedure in Algorithm 8.3. Finding all the labels with $\widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x})$ exceeding a given threshold τ can be used to create a classifier for some classic multi-label metrics, such as the Hamming loss, or to create a sparse representation of marginal vectors as discussed in Section 8.1. It is also easy to notice that Algorithm 8.3 can be modified to support different thresholds for different labels, allowing its direct application for prediction on label-wise utilities in the setting not constrained to prediction without a budget at k. Let us consider a vector of thresholds $[\tau_1, \ldots, \tau_m]$, each for label $j \in \{m\}$, instead of comparing with a constant threshold (line 9 of Algorithm 8.3), we compare with the lowest threshold of a label in the node's sub-tree $\widehat{\eta}_{v'}(\boldsymbol{x}) > \min_{j \in \mathrm{Labels}(v')} \tau_j$.

For finding top-k labels, one can apply uniform-cost search algorithm [Russell and Norvig, 2009]. Again, the procedure starts the tree traversal with the root node v_{root}. Every time it visits a node, all children of the node are added to a priority queue $\mathcal{Q}^{\mathrm{P}}$, from which the algorithm takes (pops) the node with the highest estimate of its conditional marginal probability $\widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x})$ to visit next. The algorithm stops once it visits k nodes with labels assigned to them, these labels are guaranteed to be the ones of the top-k $\widehat{\eta}_j(\boldsymbol{x})$. We present this procedure in Algorithm 8.4. The uniform-cost search is an exact algorithm, meaning that it guarantees to find precisely top-k labels with the highest estimates of conditional marginal probabilities $\widehat{\eta}_j(\boldsymbol{x})$. While in many practical applications, the above algorithms work in time close to logarithmic in the number of labels m, in some

Algorithm 8.3 Prediction of labels with $\widehat{\eta}_j(\boldsymbol{x})$ above threshold τ

Require: any tree structure $\mathcal{T}$, an estimator of marginal conditional probabilities of nodes $\widehat{\eta}(\boldsymbol{x}, v')$, threshold τ
Ensure: a prediction vector $\widehat{\boldsymbol{y}}$, such that $\widehat{y}_j = 1 \Leftrightarrow \widehat{\eta}_j(\boldsymbol{x}) > \tau$
$\widehat{\boldsymbol{y}} = \mathbf{0}$
initialize queue $\mathcal{Q} \leftarrow \varnothing$
$\widehat{\eta}^{\mathcal{T}}_{v_{\text{root}}}(\boldsymbol{x}) \leftarrow \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v_{\text{root}})$
$\mathcal{Q}$.add($(\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v_{\text{root}}), v_{\text{root}})$)
while $\mathcal{Q} \neq \varnothing$ **do**
    $(\widehat{\eta}^{\mathcal{T}}_v(\boldsymbol{x}), v) \leftarrow \mathcal{Q}$.pop()
    if $\text{lb}(v) \neq \varnothing$ **then** $\widehat{y}_{\text{lb}(v)} \leftarrow 1$
    for $v' \in \text{Ch}(v)$ **do**
        $\widehat{\eta}^{\mathcal{T}}_{v'}(\boldsymbol{x}) \leftarrow \widehat{\eta}^{\mathcal{T}}_v(\boldsymbol{x})\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v')$
        if $\widehat{\eta}^{\mathcal{T}}_{v'}(\boldsymbol{x}) > \tau$ **then** $\mathcal{Q}$.add($(\widehat{\eta}^{\mathcal{T}}_{v'}(\boldsymbol{x}), v')$)
return $\widehat{\boldsymbol{y}}$

cases, it may visit a huge part or even an entire tree. A more detailed analysis of PLT inference complexity can be found in [Busa-Fekete et al., 2019].

Algorithm 8.4 Prediction of labels with top-k $\widehat{\eta}_j(\boldsymbol{x})$ via uniform-cost search

Require: any tree structure $\mathcal{T}$, an estimator of marginal conditional probabilities of nodes $\widehat{\eta}(\boldsymbol{x}, v')$, number of top labels to predict k
Ensure: a prediction vector $\widehat{\boldsymbol{y}} = \text{select-top-}k(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}))$
$\widehat{\boldsymbol{y}} = \mathbf{0}$
initialize priority queue $\mathcal{Q}^{\text{P}} \leftarrow \varnothing$
$\widehat{\eta}^{\mathcal{T}}_{v_{\text{root}}}(\boldsymbol{x}) \leftarrow \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v_{\text{root}})$
$\mathcal{Q}^{\text{P}}$.add($(\widehat{\eta}^{\mathcal{T}}_{v_{\text{root}}}(\boldsymbol{x}), v_{\text{root}})$)
while $\|\widehat{\boldsymbol{y}}\|_1 < k$ **do**
    $(\widehat{\eta}^{\mathcal{T}}_v(\boldsymbol{x}), v) \leftarrow \mathcal{Q}^{\text{P}}$.pop()
    if $\text{lb}(v) \neq \varnothing$ **then** $\widehat{y}_{\text{lb}(v)} \leftarrow 1$
    for $v' \in \text{Ch}(v)$ **do**
        $\widehat{\eta}^{\mathcal{T}}_{v'}(\boldsymbol{x}) \leftarrow \widehat{\eta}^{\mathcal{T}}_v(\boldsymbol{x})\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v')$
        $\mathcal{Q}^{\text{P}}$.add($(\widehat{\eta}^{\mathcal{T}}_{v'}(\boldsymbol{x}), v')$)
return $\widehat{\boldsymbol{y}}$

To keep inference complexity constant, Prabhu et al. [2018b] propose to use a beam search algorithm instead. This algorithm works in a breadth-first manner, traversing the tree level by level. At each level, it visits all the nodes stored in the queue and constructs the queue for the next level nodes by keeping only b nodes with the highest $\widehat{\eta}^{\mathcal{T}}_v(\boldsymbol{x})$. The b is often referred to as the beam width. Once the search is finished, the algorithm selects the top k nodes with assigned labels among all visited nodes. Algorithm 8.5 presents the pseudocode of this method. For the complete tree with arity r, the number of evaluated nodes is upper bounded by $br \log_r m$. The trade-off for a better upper bound is the lack of guarantee that beam search will find the exact top labels, which may result in worse predictive performance [Jasinska-Kobus et al., 2020, Zhuo et al., 2020]. And while the average number of visited nodes is not necessarily smaller than that of uniform-cost search [Jasinska-Kobus et al., 2020], beam search is also easier to parallelize than uniform-cost, possibly allowing more efficient implementations.

Algorithm 8.5 Prediction of labels with top-k $\widehat{\eta}_j(\boldsymbol{x})$ via beam-search

Require: any tree structure $\mathcal{T}$, an estimator of marginal conditional probabilities of nodes $\widehat{\eta}(\boldsymbol{x}, v')$, number of top labels to predict k
Ensure: a prediction vector $\widehat{\boldsymbol{y}} = \text{select-top-}k(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}))$
$\widehat{\boldsymbol{y}} = \mathbf{0}$
initialize tree level queue $\mathcal{Q} \leftarrow \varnothing$
initialize a set of visited label nodes $\mathcal{L} \leftarrow \varnothing$
$\mathcal{Q}.\text{add}((\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v_{\text{root}}), v_{\text{root}}))$
while $\mathcal{Q} \neq \varnothing$ **do**
  $\mathcal{Q}' \leftarrow \varnothing$
  for $(\widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x}), v) \in \arg\text{top-}b_{\,(\widehat{\eta}_{v'}^{\mathcal{T}}(\boldsymbol{x}), v') \in \mathcal{Q}}\, \widehat{\eta}_{v'}^{\mathcal{T}}(\boldsymbol{x})$ **do**
    if $\text{lb}(v) \neq \varnothing$ **then** $\mathcal{L} \leftarrow \mathcal{L} \cup \text{lb}(v)$
    for $v' \in \text{Ch}(v)$ **do**
      $\widehat{\eta}_{v'}^{\mathcal{T}}(\boldsymbol{x}) \leftarrow \widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x}) \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v')$
      $\mathcal{Q}'.\text{add}((\widehat{\eta}_{v'}(\boldsymbol{x}), v'))$
  $\mathcal{Q} \leftarrow \mathcal{Q}'$
for $j \in \arg\text{top-}k_{\,j \in \mathcal{L}}\, \widehat{\boldsymbol{\eta}}_j(\boldsymbol{x})$ **do** $\widehat{y}_j \leftarrow 1$
return $\widehat{\boldsymbol{y}}$

The two algorithms for top-k prediction allow us to construct the optimal classifier for standard (not-weighted) instance-wise metrics at k, like precision@k, Hamming-score@k, (n)DCG@k. Wydmuch et al. [2021] introduced an A*-search-based algorithm [Pearl, 1984, Russell and Norvig, 2009] for efficiently finding the k labels with the highest expected gain of form $a_j \widehat{\eta}_j(\boldsymbol{x})$, where $a_j \in [0, \infty)$ is the weight given to label j. Here, we present a more general variant of that algorithm for calculating efficiently the select-top-$k(\boldsymbol{a} \odot \widehat{\boldsymbol{\eta}}(\boldsymbol{x}) + \boldsymbol{b})$, with $\in \mathbb{R}_+^m$, which is a form of optimal classifier for weighted instance-wise utilities (Section 3.2), as well as a form of classifiers forming a randomized classifier that is a result of the Frank-Wolfe algorithm (Section 7.3).

Wydmuch et al. [2021] showed that the problem of finding the label j with a node with the highest $a_j \widehat{\eta}_j(\boldsymbol{x}) = a_j \prod_{v' \in \text{Path}(v)} \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v')$ is identical to finding a node with the lowest value of $-\log a_j - \sum_{v' \in \text{Path}(v)} \log \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v')$. This makes the problem fit into the standard framework of A*-search algorithm that looks for the path with a minimal sum of costs. However, $a_j \widehat{\eta}_j(\boldsymbol{x}) + b_j$ can no longer be easily expressed in such an additive form. Because of that, we generalize this approach to the BF* (best first star)-search [Pearl, 1984] procedure that allows the use of more arbitrary functions than A*-search, being its generalization. Our algorithm resembles the already introduced uniform-cost search. It traverses a tree, starting with the tree root v_{root}. As in the case of the uniform-cost search, it uses a priority queue to decide on the order of visiting nodes. However, instead of using pure conditional probabilities of nodes $\widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x})$, the search is guided with the heuristic function $f(\boldsymbol{x}, v)$, that estimates the gain of reaching the best label node in a subtree of node v. Specifically:

$$\begin{aligned} f(\boldsymbol{x}, v) &= a_v^{\max} \widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x}) + b_v^{\max} \\ &= a_v^{\max} \prod_{v' \in \text{Path}(v)} \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v') + b_v^{\max}, \end{aligned} \tag{8.7}$$

where $a_v^{\max} = \max_{j \in \text{Labels}(v)} a_j$ and $b_v^{\max} = \max_{j \in \text{Labels}(v)} b_j$ are the largest values of a_j and b_j that belong to a label assigned to the subtree of node v. We present this procedure in Algorithm 8.6.

Algorithm 8.6 Prediction of labels with top-k $\widehat{\eta}_j^{\mathcal{T}}(\boldsymbol{x})$ via $BF^\star$-search

Require: any tree structure $\mathcal{T}$, an estimator of marginal conditional probabilities of nodes $\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, \cdot)$, number of top labels to predict k
Ensure: a prediction vector $\widehat{\boldsymbol{y}} = \text{select-top-}k(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}))$

$\widehat{\boldsymbol{y}} = \mathbf{0}$
initialize priority queue $\mathcal{Q}^{\text{P}} \leftarrow \varnothing$
$\widehat{\eta}_{v_{\text{root}}}^{\mathcal{T}}(\boldsymbol{x}) \leftarrow \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v_{\text{root}})$
$f(\boldsymbol{x}, v_{\text{root}}) \leftarrow a_{v_{\text{root}}}^{\max} \widehat{\eta}_{v_{\text{root}}}^{\mathcal{T}}(\boldsymbol{x}) + b_{v_{\text{root}}}^{\max}$
$\mathcal{Q}^{\text{P}}.\text{add}((f(\boldsymbol{x}, v_{\text{root}}), \widehat{\eta}_{v_{\text{root}}}^{\mathcal{T}}(\boldsymbol{x}), v_{\text{root}}))$
while $\|\widehat{\boldsymbol{y}}\|_1 < k$ **do**
    $(\cdot, \widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x}), v) \leftarrow \mathcal{Q}^{\text{P}}.\text{pop}()$
    if $\text{lb}(v) \neq \varnothing$ **then** $\widehat{y}_{\text{lb}(v)} \leftarrow 1$
    for $v' \in \text{Ch}(v)$ **do**
        $\widehat{\eta}_{v'}^{\mathcal{T}}(\boldsymbol{x}) \leftarrow \widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x}) \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v')$
        $f(\boldsymbol{x}, v') \leftarrow a_{v'}^{\max} \widehat{\eta}_{v'}^{\mathcal{T}}(\boldsymbol{x}) + b_{v'}^{\max}$
        $\mathcal{Q}^{\text{P}}.\text{add}((f(\boldsymbol{x}, v'), \widehat{\eta}_{v'}^{\mathcal{T}}(\boldsymbol{x}), v'))$
return $\widehat{\boldsymbol{y}}$

To guarantee that the proposed search algorithm finds the optimal solution, the top-k labels with the highest overall gains in the tree, we need to ensure that $f(\boldsymbol{x}, v)$ is admissible, i.e., it never underestimates the gain of reaching a leaf node [Pearl, 1984].[2] We also would like $f(\boldsymbol{x}, v)$ to be consistent, making the BF*-search optimally efficient, i.e., there is no other algorithm used with the heuristic that expands fewer nodes in the tree [Pearl, 1984][3].

Theorem 8.2.1. *For any $T, \widehat{\eta}(\boldsymbol{x}, \cdot), \boldsymbol{a}, \boldsymbol{b}$, and $\boldsymbol{x}$, the Algorithm 8.6 is admissible and optimally efficient.*

Proof. BF*-search finds an optimal solution if the function f is admissible, i.e., if it never underestimates the true value of $f^\star(\boldsymbol{x}, v) = \max_{j \in \boldsymbol{y}(v)} a_j \widehat{\eta}_j(\boldsymbol{x}) + b_j = a_{j^\star} \widehat{\eta}_{j^\star}(\boldsymbol{x}) + b_{j^\star}$, that is the gain of reaching the best label $j^\star$ in a subtree of node v. Since $\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, \cdot) \in [0, 1]$ and therefore $\widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x}) \geqslant \widehat{\eta}_{j^\star}(\boldsymbol{x})$, additionally $a_v^{\max} \geqslant a_{j^\star}$ and $b_v^{\max} \geqslant b_{j^\star}$ for all $v \in \mathcal{V}$, we have that $f(\boldsymbol{x}, v) \geqslant f^\star(\boldsymbol{x}, v)$, for all $v \in \mathcal{V}$, which proves admissibility.

BF*-search is optimally efficient if $f(\boldsymbol{x}, v)$ is consistent (monotone), i.e., its value is always greater than or equal to the value of $f(\boldsymbol{x}, v')$ for any of its child $v' \in \text{Ch}(v)$. Since $\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v) \in [0, 1]$ for all $v \in \mathcal{V}$, $\widehat{\eta}_{v'}^{\mathcal{T}}(\boldsymbol{x}) = \widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x}) \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v') \leqslant \widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x})$. Similar inequalities apply to the values of $a_v^{\max}$ and $b_v^{\max}$. Since $\text{Labels}(v') \subset$

[2] Both Pearl [1984], Russell and Norvig [2009] presents the problem of A*and BF*-search as a problem of minimizing the cost of reaching the goal. In their inverse definition of the problem, the function f is admissible when it never overestimates the cost of reaching the goal.

[3] These two condition are sufficient in the case of tree search, for graph search, additional properties are required for heuristic f, see [Pearl, 1984, Chapter 3] for detailed discussion on the theoretical properties of A*- and BF*-search algorithms.

Labels(v) for all $v' \in \mathrm{Ch}(v)$ it holds that:

$$a_{v'}^{\max} = \max_{j \in \mathrm{Labels}(v')} a_j \leqslant \max_{j \in \mathrm{Labels}(v)} a_j = a_v^{\max} \tag{8.8}$$

for all $v' \in \mathrm{Ch}(v)$ and with the same being true for $b_v^{\max}$. With all these inequalities combined, it holds that

$$f(\boldsymbol{x}, v') = a_{v'}^{\max} \widehat{\eta}_{v'}^{\mathcal{T}}(\boldsymbol{x}) + b_{v'}^{\max} \leqslant a_v^{\max} \widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x}) + b_v^{\max} = f(\boldsymbol{x}, v) \tag{8.9}$$

for $v' \in \mathrm{Ch}(v)$. □

The same function $f(\boldsymbol{x}, v)$ can be used with the beam search procedure (Algorithm 8.5). Of course, in this case, the guarantees no longer hold. In Section 9.4, we empirically evaluate this approach.

8.2.4 Theoretical guarantees of PLT

In this section, we briefly show that PLTs obey strong theoretical guarantees, and we are able to upper bound the L_1 estimation error of conditional marginal probabilities $|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})|$. This makes PLT suitable as a label probability estimator (LPE) in classifiers presented in Chapters 3, 6 and 7, as their guarantees rely on L_1 estimator error of the used LPE.

We start with a bound that expresses the quality of probability estimates in tree nodes $\widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x})$.

Lemma 8.2.2. *For any tree $\mathcal{T}$ and distribution $\boldsymbol{y} \sim \mathbb{P}[\boldsymbol{y} \mid \boldsymbol{x}]$ the following holds for all $v \in \mathcal{V}$:*

$$\left|\eta_v^{\mathcal{T}}(\boldsymbol{x}) - \widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x})\right| \leqslant \sum_{v' \in \mathrm{Path}(v)} \eta_{\mathrm{pa}(v')}^{\mathcal{T}}(\boldsymbol{x}) \left|\eta^{\mathcal{T}}(\boldsymbol{x}, v') - \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v')\right| , \tag{8.10}$$

where we assume $\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v) \in [0, 1]$, for each $v \in \mathcal{V}$, and $\eta_{\mathrm{pa}(v_{\mathrm{root}})}^{\mathcal{T}}(\boldsymbol{x}) = 1$.

From this lemma, we immediately get guarantees for estimates of the marginal probabilities for each label $j \in [m]$, as every label is assigned to a single node in the tree:

$$\left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| = \left|\eta_{\mathrm{v}(j)}^{\mathcal{T}}(\boldsymbol{x}) - \widehat{\eta}_{\mathrm{v}(j)}^{\mathcal{T}}(\boldsymbol{x})\right| . \tag{8.11}$$

The above result generalizes a similar result obtained for multi-class classification in [Beygelzimer et al., 2009b]. However, our bounds are tighter since the L_1 estimation error of the node classifiers is additionally multiplied by the probability of the parent node $\eta_{\mathrm{pa}(v')}^{\mathcal{T}}(\boldsymbol{x})$. The results are also obtained in a different way.

Proof. Recall the recursive factorization of probability $\eta_v^{\mathcal{T}}(\boldsymbol{x})$ given in (8.3):

$$\eta_v^{\mathcal{T}}(\boldsymbol{x}) = \eta_{\mathrm{pa}(v)}^{\mathcal{T}}(\boldsymbol{x}) \eta^{\mathcal{T}}(\boldsymbol{x}, v) .$$

As the same recursive relation holds for $\widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x})$, we have that

$$\left|\eta_v^{\mathcal{T}}(\boldsymbol{x}) - \widehat{\eta}_v^{\mathcal{T}}(\boldsymbol{x})\right| = \left|\eta^{\mathcal{T}}(\boldsymbol{x}, v) \eta_{\mathrm{pa}(v)}^{\mathcal{T}}(\boldsymbol{x}) - \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v) \widehat{\eta}_{\mathrm{pa}(v)}^{\mathcal{T}}(\boldsymbol{x})\right| . \tag{8.12}$$

By adding and subtracting $\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v)\eta^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x})$, using the triangle inequality $|a+b| \leqslant |a| + |b|$ and the assumption that $\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v) \in [0, 1]$, we obtain:

$$\begin{aligned}
\left|\eta^{\mathcal{T}}_v(\boldsymbol{x}) - \widehat{\eta}^{\mathcal{T}}_v(\boldsymbol{x})\right| &= \Big|\eta^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x})\eta^{\mathcal{T}}(\boldsymbol{x}, v) - \eta^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x})\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v) \\
&\quad + \eta^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x})\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v) - \widehat{\eta}^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x})\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v)\Big| \\
&\leqslant \left|\eta^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x})\eta^{\mathcal{T}}(\boldsymbol{x}, v) - \eta^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x})\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v)\right| \\
&\quad + \left|\eta^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x})\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v) - \widehat{\eta}^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x})\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v)\right| \\
&\leqslant \eta^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x})\left|\eta^{\mathcal{T}}(\boldsymbol{x}, v) - \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v)\right| + \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v)\left|\eta^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x}) - \widehat{\eta}^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x})\right| \\
&\leqslant \eta^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x})\left|\eta^{\mathcal{T}}(\boldsymbol{x}, v) - \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v)\right| + \left|\eta^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x}) - \widehat{\eta}^{\mathcal{T}}_{\mathrm{pa}(v)}(\boldsymbol{x})\right|.
\end{aligned} \tag{8.13}$$

Since the rightmost term corresponds to the L_1 error of the parent of v, we use recursion to get the result of Lemma 8.2.2:

$$\left|\eta^{\mathcal{T}}_v(\boldsymbol{x}) - \widehat{\eta}^{\mathcal{T}}_v(\boldsymbol{x})\right| \leqslant \sum_{v' \in \mathrm{Path}(v)} \eta^{\mathcal{T}}_{\mathrm{pa}(v')}(\boldsymbol{x})\left|\eta^{\mathcal{T}}(\boldsymbol{x}, v') - \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v')\right|, \tag{8.14}$$

where for the root node $\eta^{\mathcal{T}}_{\mathrm{pa}(v_{\mathrm{root}})}(\boldsymbol{x}) = 1$ □

The naturally arising question is whether we can bound a tree node estimation error $|\eta^{\mathcal{T}}(\boldsymbol{x}, v') - \widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v')|$. To address it, we next relate the quality of the probability estimates to the learning algorithm used to train node estimators $\widehat{\eta}^{\mathcal{T}}(\boldsymbol{x}, v')$. Precisely, if the surrogate loss ℓ used to train node estimators is a strongly proper composite loss(e.g., logistic, squared loss, squared hinge loss, or exponential) characterized by a constant λ (e.g., $\lambda = 4$ for logistic loss). For a short introduction to the strongly proper composite loss function, refer to the frame below. Then, we can express the bound (8.10) in terms of the regret of this loss function.

Theorem 8.2.3. *For any distribution $\mathbb{P}[\boldsymbol{x}, \boldsymbol{y}]$, any tree structure $\mathcal{T}$, the following holds for all $v \in \mathcal{V}$:*

$$\left|\eta^{\mathcal{T}}_v(\boldsymbol{x}) - \widehat{\eta}^{\mathcal{T}}_v(\boldsymbol{x})\right| \leqslant \sum_{v' \in \mathrm{Path}(v)} \eta^{\mathcal{T}}_{\mathrm{pa}(v')}\sqrt{\frac{2}{\lambda}}\sqrt{\mathrm{Reg}_\ell(f_{v'} \,|\, z_{\mathrm{pa}(v')} = 1, \boldsymbol{x})}, \tag{8.15}$$

where $\mathrm{Reg}_\ell(f_{v'} \,|\, z_{\mathrm{pa}(v')} = 1, \boldsymbol{x})$ is a binary classification regret for a strongly proper composite loss ℓ and λ is a constant specific for loss ℓ.

Proof. The $(\boldsymbol{x}, z_i)$ pairs are generated i.i.d. according to $\mathbb{P}[\boldsymbol{x}, z_v \,|\, z_{\mathrm{pa}(v)}]$. Assume that a node classifier has the form of a real-valued function f_v. Moreover, there exists a strictly increasing (and therefore invertible) link function $\zeta : [0, 1] \to \mathbb{R}$ such that $f_v(\boldsymbol{x}) = \zeta(\mathbb{P}[z_v \,|\, z_{\mathrm{pa}(v)}, \boldsymbol{x}])$ and ζ^{-1} is its inverse function. Recall that the regret of f_v in terms of a loss function ℓ at point $\boldsymbol{x}$ is defined as:

$$\mathrm{Reg}_\ell(f_v \,|\, z_{\mathrm{pa}(v)}, \boldsymbol{x}) = \Phi_\ell(f_v \,|\, z_{\mathrm{pa}(v)}, \boldsymbol{x}) - \Phi^\star_\ell(\boldsymbol{z}^{i-1}, \boldsymbol{x}), \tag{8.16}$$

where $\Phi_\ell(f_v \,|\, z_{\mathrm{pa}(v)} = 1, \boldsymbol{x})$ is the expected loss at point $\boldsymbol{x}$:

$$\Phi_\ell(f_v \,|\, z_{\mathrm{pa}(v)} = 1, \boldsymbol{x}) = \mathbb{P}[z_v \,|\, z_{\mathrm{pa}(v)}, \boldsymbol{x})]\ell(1, f_v(\boldsymbol{x})) + (1 - \mathbb{P}[z_v \,|\, z_{\mathrm{pa}(v)}, \boldsymbol{x})]\ell(-1, f_v(\boldsymbol{x}))\,, \tag{8.17}$$

and $\Phi^\star_\ell(\boldsymbol{x})$ is the minimum expected loss at point $\boldsymbol{x}$.

If a node classifier is trained by a learning algorithm that minimizes a strongly proper composite loss, then the bound (8.10) can be expressed in terms of the regret of this loss function [Agarwal, 2014]:

$$\left|\mathbb{P}[z_v \,|\, z_{\mathrm{pa}(v)}, \boldsymbol{x})] - \zeta^{-1}(f_v)\right| \leqslant \sqrt{\frac{2}{\lambda}}\sqrt{\mathrm{Reg}_\ell(f_v \,|\, z_{\mathrm{pa}(v)}, \boldsymbol{x})}\,. \tag{8.18}$$

By putting the above inequality into (8.10), we get

$$\begin{aligned}
\left|\eta^{\mathcal{T}}_v(\boldsymbol{x}, j) - \eta^{\mathcal{T}}_v(\boldsymbol{x})\right| &\leqslant \sum_{v' \in \mathrm{Path}(v)} \mathbb{P}[z_{\mathrm{pa}(v')} \,|\, \boldsymbol{x}] \left|\mathbb{P}[z_{v'} \,|\, z_{\mathrm{pa}(v')}, \boldsymbol{x})] - \widehat{\mathbb{P}}[z_i \,|\, z_{\mathrm{pa}(v)}, \boldsymbol{x}]\right| \\
&= \sum_{v' \in \mathrm{Path}(v)} \mathbb{P}[z_{\mathrm{pa}(v')} \,|\, \boldsymbol{x}] \left|\mathbb{P}[z_{v'} \,|\, z_{\mathrm{pa}(v')}, \boldsymbol{x})] - \zeta^{-1}(f_{v'})\right| \\
&\leqslant \sum_{v' \in \mathrm{Path}(v)} \mathbb{P}[z_{\mathrm{pa}(v')} \,|\, \boldsymbol{x}] \sqrt{\frac{2}{\lambda}}\sqrt{\mathrm{Reg}_\ell(f_{v'} \,|\, z_{\mathrm{pa}(v')}, \boldsymbol{x})}
\end{aligned} \tag{8.19}$$

□

The above result shows that the absolute error of estimating the marginal probability of label j can be upper-bounded by the regret of the node classifiers on the corresponding path from the root to a label's node. Moreover, this result shows that for zero-regret (i.e., optimal) node probability estimates, we obtain an optimal multi-label estimator in terms of marginal conditional probabilities $\eta^{\mathcal{T}}_j(\boldsymbol{x})$. An exhaustive discussion on theoretical guarantees of PLTs can be found in [Busa-Fekete et al., 2019, Jasinska-Kobus et al., 2020].

Strongly proper composite losses

The strongly proper composite losses are of special interest in the problem of class probability estimation with two outcomes, $y \in \{0, 1\}$. Let pairs $(\boldsymbol{x}, y)$ be generated i.i.d. according to $\mathbb{P}[\boldsymbol{x}, y]$. With $\mathbb{P}[y = 1 \,|\, \boldsymbol{x}]$ being denoted by $\eta(\boldsymbol{x})$ and its estimate by $\widehat{\eta}(\boldsymbol{x}) \in [0, 1]$. A label probability estimation (LPE) loss function $\ell : \{0, 1\} \times [0, 1] \mapsto \mathbb{R}_+$, with its conditional risk is given by

$$\Phi_\ell(\widehat{\eta} \,|\, \boldsymbol{x}) = \eta(\boldsymbol{x})\ell(1, \widehat{\eta}(\boldsymbol{x})) + (1 - \eta(\boldsymbol{x}))\ell(-1, \widehat{\eta}(\boldsymbol{x}))\,. \tag{8.20}$$

A LPE loss is proper if for any $\eta(\boldsymbol{x}) \in [0, 1]$, $\eta(\boldsymbol{x}) \in \arg\min_{\widehat{\eta}} R_\ell(\widehat{\eta} \,|\, \boldsymbol{x})$. Since it is often more convenient for prediction algorithms to work with a real-valued scoring function, $f : \mathcal{X} \mapsto \mathbb{R}$, then with an estimate bounded to interval $[0, 1]$, we transform $\widehat{\eta}(\boldsymbol{x})$ using a strictly increasing (and therefore invertible) link function $\zeta : [0, 1] \to \mathbb{R}$, that is, $f(\boldsymbol{x}) = \zeta(\widehat{\eta}(\boldsymbol{x}))$. Let us consider a composite

loss function $\ell_c : \{0,1\} \times \mathbb{R} \mapsto \mathbb{R}_+$ defined via LPE loss as

$$\ell_c(y, f(\boldsymbol{x})) = \ell(y, \zeta^{-1}(f(\boldsymbol{x}))) \,. \tag{8.21}$$

The regret of f in terms of a loss function ℓ_c at point $\boldsymbol{x}$ is defined as:

$$\mathrm{Reg}_{\ell_c}(f \,|\, \boldsymbol{x}) = \Phi_\ell(\zeta^{-1}(f) \,|\, \boldsymbol{x}) - \Phi^\star_\ell(\boldsymbol{x}) \,, \tag{8.22}$$

where $\Phi^\star_\ell(\boldsymbol{x})$ is the minimum expected loss at point $\boldsymbol{x}$, achievable by $f^\star(\boldsymbol{x}) = \zeta(\eta(\boldsymbol{x}))$.

It is said that loss function ℓ_c is λ-strongly proper composite loss, if for any $\eta(\boldsymbol{x}), \zeta^{-1}(f(\boldsymbol{x})) \in [0,1]$:

$$\left|\eta(\boldsymbol{x}) - \zeta^{-1}(f(\boldsymbol{x}))\right| \leqslant \sqrt{\frac{2}{\lambda}} \sqrt{\mathrm{Reg}_{\ell_c}(f \,|\, \boldsymbol{x})} \,. \tag{8.23}$$

It can be shown under mild regularity assumptions that ℓ_c is λ-strongly proper composite if and only if its corresponding LPE loss is proper and function $H_\ell(\eta) = \Phi_\ell(\eta \,|\, \boldsymbol{x})$ is λ-strongly concave, that is, $\left|\frac{d^2 H_\ell(\eta)}{d^2 \eta}\right| \geqslant \lambda$.

For a more detailed introduction to strongly proper composite losses, we refer the reader to [Agarwal, 2014].

8.3 Summary of the chapter

In this chapter, first, we have demonstrated how leveraging sparse representations of $\widehat{\boldsymbol{\eta}}(\boldsymbol{x})$ can significantly reduce the computational complexity of the algorithms discussed in previous chapters. When the sparse representation preserves enough of the highest probability values, the increase in regret is minimal. Moreover, this sparsity approach can be easily applied to nearly any label probability estimator (LPE) used in XMLC.

Next, we have discussed probabilistic label trees (PLTs), a popular family of LPEs in XMLC that offer reduced complexity in both training and prediction. We have shown how they can be employed as outputs of neural networks. Additionally, we have introduced an A*-search algorithm that, in contrast to the sparsity technique, allows for exact efficient top-k prediction under general instance-wise weighted utilities. Finally, we demonstrated that PLTs obey strong theoretical guarantees when trained with strictly proper losses. Specifically, we have derived a bound on the L_1 estimation error of marginal conditional probabilities of labels, which is an essential component of the bounds we provided for inference algorithms introduced in the previous chapters.

9 Experiments

In this chapter, we conduct a series of empirical experiments in order to validate the theoretical findings presented in this thesis and to compare the introduced inference algorithms. These experiments can be treated as a replication and unification under the same setup of the experiments initially performed in [Wydmuch et al., 2021, Schultheis et al., 2023, 2024].

9.1 General experimental setup

We start with a description of datasets, utilities, and the form of Label Probability Estimator (LPE) used in all experiments described in this chapter. To empirically test the introduced inference algorithms, we use nine popular benchmarks from the XMLC repository [Bhatia et al., 2016] covering a wide range of data and label set sizes, starting with a dataset with over a few thousand labels and ending with a few hundred thousand labels. The list of datasets and their statistics is presented in Table 9.1.

In our experiments, we optimize and evaluate algorithms under three instance-wise utilities:

- Instance-P@k – the standard precision@k (as defined in (3.1)),
- Instance-PSP@k – the propensity-scored precision@k (as defined in (4.12)), without normalization, with propensities coming from the empirical propensity model of Jain et al. [2016] (as defined in (4.16)), with parameters for each dataset as recommended by the authors,
- Instance-R@k – the standard recall@k (as defined in (3.18)),

and six macro-averaged utilities:

- Macro-BA@k – macro-balanced accuracy@k (as defined in Table 5.1).
- Macro-P@k – macro-precision@k (as defined in Table 5.1),

Table 9.1: The number of unique labels, observations in train and test splits, the average number of true labels per sample ($\mathbb{E}\left[|\boldsymbol{y}|_1\right]$), the average number of positive instances per label j ($\mathbb{E}\left[\pi_j n_j\right]$), and an inter-label imbalance ratio $\left(\text{ILIR} := \frac{\max\{\pi_i : i \in [m]\}}{\min\{\pi_j : j \in [m]\}}\right)$ in the benchmark data.

Dataset	m	n_{train}	n_{test}	$\mathbb{E}\left[\lvert\boldsymbol{y}\rvert_1\right]$	$\mathbb{E}\left[\pi_j n_j\right]$	ILIR
RCV1x-2K	2456	623847	155962	4.76	1510.13	13799
EURLex-4K	3993	15539	3809	5.31	25.97	1253
EURLex-4.3K	4271	45000	6,000	5.07	61.69	4146
AmazonCat-13K	13330	1186239	306782	5.05	566.01	223441
AmazonCat-14K	14588	4398050	1099725	3.53	1330.10	560695
Wiki10-31K	30938	14146	6616	18.76	12.59	8378
WikiLSHTC-325K	325056	1778351	587084	3.26	23.74	387168
WikipediaLarge-500K	501070	1813391	783743	4.77	24.75	199677
Amazon-670K	670091	490449	153025	5.38	5.17	2258

- Macro-R@k – macro-recall@k (as defined in Table 5.1),
- Macro-F_1@k – macro-F_1-measure@k (as defined in Table 5.1),
- Macro-JS@k – macro-Jaccard similarity@k (as defined in Table 5.1),
- Cov@k – (macro-)coverage@k (as defined in Table 5.1).

To obtain a label probability estimator (LPE), we use an ensemble of probabilistic label trees. We use the napkinXC [Jasinska-Kobus et al., 2020] library to train an ensemble of three PLTs with different tree structures, each built via a popular approach of hierarchically clustering labels [Wydmuch et al., 2018, Prabhu et al., 2018b, You et al., 2019a, Yu et al., 2022], until all clusters reach a size of 400 or less. The tree nodes are then learned using LIBLINEAR [Fan et al., 2008] with L2-regularized logistic loss. We represent instances using sparse TF-IDF features; thanks to that, we cover a large part of datasets from the XMLC repository, as some of them are only available in such a form.

For all datasets, we train models using the suggested hyperparameters on a training set (we report the details for training in Appendix B.1) and then use them to predict $\widehat{\boldsymbol{\eta}}(\boldsymbol{x})$ for all instances in a validation set and test set. These estimates were then plugged into different inference strategies. To run the inference algorithm efficiently for XMLC datasets, we pre-select for each instance the top $k' = 100$ labels with the highest $\widehat{\eta}_j(\boldsymbol{x})$ as described in Section 8.1, 100 was found to be a good trade-off between speed and predictive performance by Schultheis et al. [2023].

Using these predictions, we compare the following inference methods:

- Top-k – selects k labels with the highest $\widehat{\eta}_j(\boldsymbol{x})$. This is the optimal strategy for precision@k and nDCG@k, and also the default prediction strategy in many XMLC methods.
- PS-k – selects k labels with the highest $p_j \widehat{\eta}_j(\boldsymbol{x})$, with p_j given by the empirical model of Jain et al. [2016] (4.16), this is the optimal strategy for propensity-scored precision@k with the same p_j.
- Pow-k – selects k labels with the highest $g_j^{\text{pl}} \widehat{\eta}_j(\boldsymbol{x})$, where $g_j^{\text{pl}} = \pi_j^{-\beta}$ is the popular power-law-based weighting (4.19), we report results for $\beta \in$

$\{0.25, 0.5\}$.

- Log-k – selects k labels with the highest $g_j^{\log}\widehat{\eta}_j(\boldsymbol{x})$, where $g_j^{\log} = -\log \pi_j$ is the logarithmic weighting.
- Macro-$\mathrm{R}_{\text{prior}}$-$k$ – the optimal strategy for macro-averaged recall based on priors: selection of k labels with the highest $\frac{\eta_j(\boldsymbol{x})}{\pi_j}$ (see Section 7.2 for details).
- Macro-$\mathrm{BA}_{\text{prior}}$-$k$ – the optimal strategy for macro-averaged balanced-accuracy: selection of k labels with the highest $\left(\frac{1}{2\pi_j} + \frac{1}{2(1-\pi_j)}\right)\widehat{\eta}_j(\boldsymbol{x}) - \frac{1}{2(1-\pi_j)}$ (see Section 7.2 for details).
- BCA$(\cdot)$ – the block coordinate ascent for finding the optimal prediction under the ETU framework (Algorithm 6.1) for a given utility. We use this method to optimize all of the macro-averaged utilities listed above. To ensure the smoothness and Lipschitzness of optimized utilities, a small value υ is added to the denominator (with the exception of the coverage metric, which is not a continuous function).
- FW$(\cdot)$ – the Frank Wolfe-based algorithm for finding optimal randomized classifier under the PU framework (Algorithm 7.1) for a given utility. Similarly to BCA, we use this method for all of the macro-averaged utilities listed above, except coverage, for which optimization does not make sense under the PU framework. Again, we add a small υ to the denominator of the optimized utility.

For all methods that rely on priors $\boldsymbol{\pi}$ (PS-k, Pow-k, Log-k, Macro-$\mathrm{R}_{\text{prior}}$-$k$, and Macro-$\mathrm{BA}_{\text{prior}}$-$k$), these values are estimated using the training set.

9.2 Comparison of inference algorithms

Wydmuch et al. [2021] and Schultheis et al. [2023, 2024] evaluated discussed inference methods independently of each other. In this section, we present a unified comparison of all inference algorithms under a consistent experimental setup, that is, using the same base datasets, metrics, and label probability estimators (LPEs). Such a comparison is not trivial due to the different assumptions under which these methods operate.

The ETU framework optimizes for a specific test set but requires the whole set to be available during inference. In contrast, the PU framework allows predictions to be made independently and aims for optimality on a population level. As such, we expect the Block Coordinate Ascent algorithm to outperform the Frank-Wolfe method on finite benchmark test sets.

Another important difference between the algorithms is that the FW algorithm requires not only a probability estimator but also an additional validation set to construct a randomized classifier that can later make point-wise predictions on the test set. Ideally, this dataset should be a separate set taken from the training set to prevent overfitting, but because of data sparsity in the XMLC setting, withholding

data from LPE's training set can significantly degrade its quality and hurt the final performance more severely than overfitting the FW procedure. Because of that, it might be more beneficial to use the same data for LPN's training and FW procedure [Schultheis et al., 2024].

Lastly, the provided train and test splits in the XML repository are not random, as they have been obtained to maximize the number of unique labels in both sets [Bhatia et al., 2016]. This, again, gives the ETU algorithm a certain advantage, as it does not require the test set to follow an i.i.d. distribution.

Because of the above reason, we decided to conduct two types of experiments for each dataset:

1. Experiments with synthetic labels – in the first type, we aim to simulate the case when we have access to true marginal probabilities $\boldsymbol{\eta} = \widehat{\boldsymbol{\eta}}$. Additionally, we want the train and test sets to meet the i.i.d. assumption. Finally, we want to show that when n is growing, the results of FW get close to the results of the BCA method working in the ETU framework as ETU converges to PU (asymptotic equivalence (2.17)). To achieve these goals, for each dataset, we first merge the original train and test sets and duplicate the obtained set five times (to simulate $n \to \infty$)), with the exception of the AmazonCat-14K dataset, for which we duplicate data only two times. We then discard true labels, and in their place, we sample new labels according to $\widehat{\boldsymbol{\eta}}$ obtained from the estimator. Half of the obtained set is then used as a validation set for the FW algorithm, and the other half is used as a test set for all the methods. The same $\widehat{\boldsymbol{\eta}}$ is then also used as an input to the inference algorithms, simulating the perfect knowledge about the true η.
2. Experiments on original benchmark datasets – in the second type, we compare all the methods on the original benchmark datasets, keeping the provided training-test data splits and original labels. This ensures that our results remain comparable with the existing literature. At the same time, the experiments on the original datasets better reflect not only huge imbalances of labels but also the high sparsity of data for tail-labels.

9.2.1 Experiments on datasets with synthetic labels

We present the results on synthetic variants of benchmark datasets in Table 9.2. The results are reported for $k = 5$ as this is one of the most common values used in the literature as well as in real-world production systems. In Table B.2, we also present the results for $k = 1$ and $k = 3$. We do not report results for higher values of k because many of the benchmark datasets have an average number of positive labels close to 5, making the difference between Top-k inference and other methods much less pronounced. Due to the size of the synthetic data, for each dataset, we perform inference using each method only once, but we repeat the generation of test labels 5 times and report the average results, with standard deviations included in the tables in Appendix B.2.

For the readability of the results, we not only mark the **best** result and the *second best* result but for every method, we also highlight the cells with the

result on a metric this method aims to optimize using the green background. In addition, we gray out the results that are below the values presented in cells with the green background. In general, we expect that inference strategies that target a specific metric obtain the best results.

In these synthetic experiments, the regret of the algorithms introduced is eliminated since there is no error in estimating $\boldsymbol{\eta}(\boldsymbol{x})$ (with the exception of BCA algorithms where small regret can be caused by the additional approximations used in these algorithms). As expected, we observe that the results in green cells are indeed the best, with the small exception of methods optimizing for macro-recall@k being also the best on macro-balanced accuracy@k and vice versa due to their similar form of the optimal classifier. The same can sometimes be observed for macro-F_β-measure@k and macro-Jaccard similarity@k due to their very similar formulation. We also observed that the closed-form methods for Macro-$\mathrm{R}_{\mathrm{prior}}$-$k$ and Macro-$\mathrm{BA}_{\mathrm{prior}}$-$k$ give indeed similar results to the ones obtained using BCA or FW methods. In fact, we observe that the FW algorithm recovers a similar solution by finding a randomized classifier composed mostly by a single subclassifier with weights $\boldsymbol{a}$ and $\boldsymbol{b}$ being a scaled version of the closed form solution.

While BCA wins in all cases with the FW algorithm, the difference between both methods is very small. The only exception is macro-precision@k, for which FW achieves results lower than BCA by 1 p.p. or more in a few cases. This might be because precision is the only metric considered that is not L-Lipschitz, and even with the added constant υ, it has an extreme magnitude of gradients near 0. Because of that, the FW algorithm is prone to overfitting to the validation set. This problem will become even more visible in the next section when experiments on the original datasets are discussed.

For all synthetic experiments, we used υ = 1e-8. The stopping conditions for both BCA and FW were set to a maximum of 100 iterations, the minimal improvement in the objective to continue had to be greater than ϵ = 1e-7. Furthermore, for the FW algorithm, the contribution of a new classifier to the improvement in the objective must be greater than ϵ_α = 1e-3.

9.2.2 Experiments on original datasets

We present the results on the original benchmark datasets in Table 9.3. The results are reported for $k = 5$ as in the previous experiments. Table B.2 in Appendix B.2 contains additional results for $k = 1$ and $k = 3$. We repeat the inference using each method five times and report the averages. The standard deviations can be found in the Appendix.

In this set of experiments, we clearly see the effect of inaccurate probability estimates. Even the simple inference strategies, like Top-k, are sometimes not the best on their target measures. Still, in most cases, the BCA method is the best. However, this time, FW performs much worse than BCA in many cases, especially for macro-precision@k, but also for macro-F_β-measure@k and macro-Jaccard similarity@k on datasets with the largest number of labels. Again, we attribute this to the overfitting of the FW inference method. Note that

Table 9.2: Results (%) for $k = 5$ on synthetic versions of XMLC datasets with ideal estimates of marginal conditional probabilities $\boldsymbol{\eta}(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}(\boldsymbol{x})$. The green background indicates cells in which the inference algorithm matches the metric it optimizes and the gray text indicates results worse than those results for a given metric. The best results are in **bold**, and the second best are in *italic*. * – because in this experiment we sample labels independently, Top-k becomes the optimal strategy for recall@k as showed in Theorem 3.3.1.

Method	Instance @5			Macro @5					
	P	PS	R	P	R	BA	F_1	JS	Cov
Synthetic RCV1x-2K									
Top-5	**54.70**	*57.18*	* **76.75**	32.50	17.05	58.48	17.35	10.78	76.95
PS-5	*54.32*	**57.88**	*76.55*	38.15	45.91	72.91	36.11	23.32	99.70
Pow-$5_{\beta=0.25}$	52.66	56.90	75.37	33.83	53.57	76.74	37.38	24.25	99.70
Pow-$5_{\beta=0.5}$	49.42	54.08	72.42	25.39	62.15	81.02	31.06	19.74	*99.84*
Log-5	52.99	57.10	75.52	37.81	44.77	72.34	37.00	23.92	98.94
Macro-R_{prior}-5	36.69	41.37	56.73	17.31	68.59	84.23	19.86	12.20	**99.90**
Macro-BA_{prior}-5	37.13	41.79	57.43	17.36	68.58	84.23	19.92	12.24	**99.90**
BCA(Macro-P@5)	32.75	33.19	49.34	**74.72**	5.46	52.68	7.57	4.50	92.61
BCA(Macro-R@5)	31.98	36.47	49.79	16.98	**69.30**	**84.58**	18.40	11.28	**99.90**
BCA(Macro-BA@5)	32.66	37.13	50.92	17.02	**69.30**	**84.58**	18.46	11.33	**99.90**
BCA(Macro-F_1@5)	49.19	53.35	71.05	46.34	50.10	75.00	**46.56**	*31.35*	99.69
BCA(Macro-JS@5)	49.24	53.40	71.02	46.07	50.14	75.02	*46.47*	**31.36**	99.68
BCA(Cov@5)	1.14	2.22	0.77	11.12	44.85	72.32	4.78	2.54	**99.90**
FW(Macro-P@5)	30.63	31.08	46.21	*72.33*	6.22	53.05	7.36	4.37	89.88
FW(Macro-R@5)	31.93	36.42	49.69	16.98	*69.29*	*84.57*	18.39	11.28	**99.90**
FW(Macro-BA@5)	32.62	37.09	50.85	17.02	69.28	*84.57*	18.45	11.32	**99.90**
FW(Macro-F_1@5)	49.17	53.31	71.06	46.11	49.85	74.88	46.24	31.14	99.57
FW(Macro-JS@5)	49.22	53.37	71.00	45.89	49.88	74.89	46.21	31.19	99.54
Synthetic EURLex-4K									
Top-5	**65.85**	130.35	* **73.84**	51.16	57.70	78.83	52.42	40.40	82.77
PS-5	64.83	**134.08**	72.77	49.35	67.84	83.90	55.17	42.29	88.41
Pow-$5_{\beta=0.25}$	64.74	*133.95*	72.67	50.22	67.31	83.63	55.66	42.82	88.15
Pow-$5_{\beta=0.5}$	62.14	132.08	69.81	45.55	69.69	84.82	52.64	39.72	88.85
Log-5	*65.05*	133.80	*73.02*	51.46	66.08	83.02	56.13	43.38	87.69
Macro-R_{prior}-5	49.94	116.20	56.13	34.98	72.05	85.99	41.78	29.80	90.12
Macro-BA_{prior}-5	50.06	116.37	56.29	35.03	72.04	85.99	41.84	29.84	90.12
BCA(Macro-P@5)	20.47	36.54	22.48	**67.71**	24.95	62.44	27.13	19.45	84.04
BCA(Macro-R@5)	49.33	115.16	55.38	34.11	**72.83**	**86.38**	40.79	28.96	*90.70*
BCA(Macro-BA@5)	49.46	115.34	55.53	34.17	*72.82*	**86.38**	40.87	29.03	90.68
BCA(Macro-F_1@5)	62.65	129.11	69.84	56.06	64.88	82.42	**58.51**	**45.85**	88.66
BCA(Macro-JS@5)	62.66	129.12	69.81	56.07	64.86	82.41	*58.50*	**45.85**	88.66
BCA(Cov@5)	9.12	34.62	10.19	28.67	49.89	74.89	17.65	10.85	**91.39**
FW(Macro-P@5)	19.72	35.46	21.61	*65.70*	24.74	62.33	26.31	18.86	82.03
FW(Macro-R@5)	48.42	113.96	54.32	33.54	72.47	*86.20*	40.05	28.34	90.44
FW(Macro-BA@5)	48.54	114.13	54.46	33.60	72.47	*86.20*	40.11	28.40	90.48
FW(Macro-F_1@5)	62.61	129.04	69.80	55.94	64.69	82.32	58.27	*45.63*	88.48
FW(Macro-JS@5)	62.62	129.07	69.78	55.94	64.69	82.32	58.25	*45.63*	88.44
Synthetic EURLex-4.3K									
Top-5	**73.79**	114.28	* **82.02**	52.29	57.28	78.62	52.71	41.14	81.34
PS-5	*72.93*	**118.07**	*81.21*	53.49	75.26	87.61	60.50	47.67	90.93
Pow-$5_{\beta=0.25}$	72.53	*117.97*	80.83	53.93	75.15	87.56	60.91	48.11	90.80
Pow-$5_{\beta=0.5}$	70.68	116.64	78.94	49.28	77.53	88.75	57.61	44.74	91.40
Log-5	72.77	117.81	81.06	55.25	73.25	86.61	61.26	48.61	90.07
Macro-R_{prior}-5	61.88	106.33	69.15	38.49	79.85	89.90	46.66	34.52	92.32
Macro-BA_{prior}-5	62.04	106.52	69.33	38.55	79.85	89.90	46.72	34.57	92.32
BCA(Macro-P@5)	30.93	41.34	33.80	**70.96**	24.92	62.43	26.70	19.23	86.62
BCA(Macro-R@5)	60.15	104.09	67.17	37.05	**80.19**	**90.07**	44.89	32.99	*92.60*
BCA(Macro-BA@5)	60.32	104.29	67.38	37.10	*80.18*	**90.07**	44.95	33.04	*92.60*
BCA(Macro-F_1@5)	70.77	114.91	78.50	60.70	73.15	86.56	**64.76**	*52.36*	91.10
BCA(Macro-JS@5)	70.70	114.84	78.40	60.69	73.15	86.56	*64.75*	**52.38**	91.10
BCA(Cov@5)	5.83	19.72	6.28	22.97	53.56	76.72	16.62	10.38	**92.89**
FW(Macro-P@5)	30.84	41.04	33.69	*68.89*	24.64	62.29	26.06	18.77	84.32
FW(Macro-R@5)	60.22	104.17	67.23	36.91	80.05	*90.00*	44.69	32.84	92.52
FW(Macro-BA@5)	60.39	104.40	67.43	36.96	80.05	*90.00*	44.74	32.89	92.52
FW(Macro-F_1@5)	70.76	114.94	78.50	60.62	72.96	86.46	64.60	52.22	90.92
FW(Macro-JS@5)	70.69	114.85	78.40	60.60	72.98	86.47	64.57	52.23	90.95

Method	Instance @5			Macro @5					
	P	PS	R	P	R	BA	F_1	JS	Cov
Synthetic AmazonCat-13K									
Top-5	**63.16**	71.34	* **76.12**	45.39	47.71	73.85	42.12	30.42	82.48
PS-5	*62.63*	**72.70**	*75.72*	43.95	71.44	85.71	51.43	37.33	93.86
Pow-$5_{\beta=0.25}$	59.86	*71.35*	73.51	40.80	75.01	87.50	50.70	36.52	93.79
Pow-$5_{\beta=0.5}$	54.68	66.80	68.01	31.75	80.13	90.06	42.63	29.53	94.44
Log-5	59.92	71.18	73.44	44.13	69.47	84.73	52.08	37.99	92.03
Macro-R_{prior}-5	38.70	50.58	47.71	20.81	**83.87**	91.92	28.83	19.13	94.85
Macro-BA_{prior}-5	39.08	50.96	48.34	20.84	**83.87**	**91.92**	28.86	19.15	94.85
BCA(Macro-P@5)	33.90	34.54	42.20	**72.72**	12.27	56.12	14.55	9.20	88.37
BCA(Macro-R@5)	35.31	46.89	43.42	20.54	**84.33**	**92.15**	27.82	18.62	*95.11*
BCA(Macro-BA@5)	35.68	47.25	44.04	20.56	**84.33**	**92.15**	27.84	18.64	*95.11*
BCA(Macro-F_1@5)	57.09	67.75	69.94	52.89	66.34	83.16	**57.45**	**43.31**	93.41
BCA(Macro-JS@5)	56.98	67.65	69.76	52.87	66.35	83.17	*57.44*	**43.31**	93.41
BCA(Cov@5)	2.62	5.14	2.43	8.11	47.63	73.80	7.87	4.39	**95.13**
FW(Macro-P@5)	33.30	33.92	41.31	*71.50*	12.23	56.10	14.20	9.00	86.48
FW(Macro-R@5)	35.37	46.95	43.50	20.50	*84.29*	*92.13*	27.76	18.57	95.08
FW(Macro-BA@5)	35.74	47.31	44.13	20.52	*84.29*	*92.13*	27.78	18.59	95.08
FW(Macro-F_1@5)	57.07	67.75	69.92	52.90	66.24	83.11	**57.37**	*43.27*	93.28
FW(Macro-JS@5)	56.94	67.62	69.72	52.86	66.24	83.11	57.35	**43.26**	93.27
Synthetic AmazonCat-14K									
Top-5	**52.43**	60.15	* **83.00**	37.63	50.29	75.13	40.68	29.11	81.49
PS-5	*52.19*	**60.67**	*82.73*	33.47	67.54	83.76	42.04	29.44	92.44
Pow-$5_{\beta=0.25}$	50.84	59.90	81.10	30.73	70.56	85.27	40.55	28.03	92.41
Pow-$5_{\beta=0.5}$	45.94	55.54	74.21	22.26	76.95	88.47	31.81	21.10	93.18
Log-5	51.45	*60.22*	81.85	34.67	63.84	81.91	43.04	30.29	90.11
Macro-R_{prior}-5	29.69	39.03	44.01	14.93	**81.32**	90.65	21.33	13.78	93.63
Macro-BA_{prior}-5	29.92	39.26	44.84	14.94	**81.32**	**90.65**	21.35	13.79	93.63
BCA(Macro-P@5)	25.93	26.70	47.71	**72.16**	8.24	54.11	11.13	7.15	82.25
BCA(Macro-R@5)	26.50	35.59	36.55	14.90	**81.95**	**90.96**	20.75	13.52	*94.00*
BCA(Macro-BA@5)	29.48	38.66	43.27	15.24	*81.93*	*90.95*	21.28	13.91	*94.00*
BCA(Macro-F_1@5)	48.73	56.60	77.80	48.69	57.31	78.65	**51.58**	**37.86**	91.22
BCA(Macro-JS@5)	48.60	56.44	77.54	48.74	57.27	78.63	**51.58**	**37.86**	91.22
BCA(Cov@5)	1.36	3.29	1.72	7.76	46.83	73.40	5.20	2.84	**94.08**
FW(Macro-P@5)	25.83	26.60	47.52	*65.54*	8.15	54.07	10.28	6.57	74.90
FW(Macro-R@5)	26.50	35.58	36.56	14.86	**81.86**	90.92	20.71	13.49	93.95
FW(Macro-BA@5)	29.49	38.66	43.36	15.20	**81.84**	**90.91**	21.23	13.88	93.95
FW(Macro-F_1@5)	48.72	56.59	77.79	48.62	57.02	78.50	*51.33*	*37.67*	90.85
FW(Macro-JS@5)	48.62	56.46	77.58	48.67	56.98	78.48	51.32	*37.67*	90.85
Synthetic Wiki10-31K									
Top-5	**80.42**	107.38	* **36.82**	11.58	4.47	52.23	5.77	3.94	14.98
PS-5	*65.29*	**245.24**	*30.97*	58.14	68.79	84.39	58.06	46.82	89.24
Pow-$5_{\beta=0.25}$	64.33	*244.46*	30.22	57.97	67.85	83.92	57.82	46.63	88.63
Pow-$5_{\beta=0.5}$	57.78	240.67	26.07	55.13	69.86	84.93	57.01	45.55	89.73
Log-5	62.97	237.25	28.16	57.02	63.46	81.73	55.63	44.77	85.45
Macro-R_{prior}-5	48.22	228.66	19.74	49.34	**70.47**	85.23	53.21	41.54	89.97
Macro-BA_{prior}-5	48.34	228.80	19.84	49.38	**70.47**	**85.23**	53.22	41.55	89.98
BCA(Macro-P@5)	54.89	136.56	24.32	**65.09**	41.01	70.50	42.85	32.79	84.99
BCA(Macro-R@5)	46.84	223.63	18.89	48.61	**72.24**	**86.11**	53.43	41.48	91.65
BCA(Macro-BA@5)	46.91	223.72	18.94	48.63	**72.24**	**86.11**	53.43	41.48	91.66
BCA(Macro-F_1@5)	58.45	234.90	26.30	59.24	69.65	84.82	**60.44**	*48.54*	*92.10*
BCA(Macro-JS@5)	58.03	235.73	26.04	58.81	70.17	85.08	*60.31*	**48.58**	92.00
BCA(Cov@5)	39.75	200.86	15.32	45.30	69.04	84.52	49.05	37.02	**92.59**
FW(Macro-P@5)	55.28	137.62	24.48	*62.12*	40.24	70.12	41.41	31.77	81.43
FW(Macro-R@5)	46.87	225.21	18.90	48.68	*71.05*	*85.52*	53.02	41.23	90.79
FW(Macro-BA@5)	46.95	225.30	18.96	48.71	*71.05*	*85.52*	53.03	41.24	90.80
FW(Macro-F_1@5)	58.47	236.58	26.30	58.99	68.19	84.09	**59.18**	47.50	90.87
FW(Macro-JS@5)	58.10	237.31	26.07	58.65	68.96	84.48	59.30	**47.75**	90.85

Method	Instance @5			Macro @5					
	P	PS	R	P	R	BA	F_1	JS	Cov
Synthetic WikiLSHTC-325K									
Top-5	**30.62**	66.19	* **66.39**	29.00	60.89	80.44	36.46	25.57	79.09
PS-5	*30.07*	**67.97**	*65.75*	27.15	69.14	84.57	36.46	25.33	83.92
Pow-$5_{\beta=0.25}$	29.36	*67.60*	64.69	27.56	68.90	84.45	36.93	25.75	83.68
Pow-$5_{\beta=0.5}$	27.18	65.76	60.37	23.85	70.52	85.26	33.19	22.61	84.19
Log-5	29.84	67.36	65.38	28.97	65.83	82.91	37.72	26.50	82.10
Macro-R_{prior}-5	24.42	62.44	55.21	19.39	71.55	85.77	27.81	18.40	84.44
Macro-BA_{prior}-5	24.42	62.44	55.21	19.39	71.55	85.77	27.81	18.40	84.44
BCA(Macro-P@5)	12.48	22.64	25.90	**50.21**	29.82	64.91	27.75	19.73	75.65
BCA(Macro-R@5)	22.11	58.75	50.63	17.37	**72.14**	**86.07**	24.61	16.08	*84.99*
BCA(Macro-BA@5)	22.11	58.75	50.63	17.37	**72.14**	**86.07**	24.61	16.08	*84.99*
BCA(Macro-F_1@5)	26.83	60.44	58.14	40.93	61.81	80.90	**47.11**	**34.87**	83.00
BCA(Macro-JS@5)	26.80	60.43	58.04	40.92	61.81	80.91	*47.10*	**34.87**	82.99
BCA(Cov@5)	5.40	21.92	11.72	17.97	56.52	78.26	16.81	10.53	**85.33**
FW(Macro-P@5)	12.49	22.68	25.88	*49.06*	29.52	64.76	27.07	19.28	73.41
FW(Macro-R@5)	22.11	58.79	50.63	17.14	*71.95*	*85.97*	24.23	15.84	84.83
FW(Macro-BA@5)	22.11	58.79	50.63	17.14	*71.95*	*85.97*	24.23	15.84	84.83
FW(Macro-F_1@5)	26.83	60.42	58.13	40.88	61.40	80.70	46.84	*34.68*	82.65
FW(Macro-JS@5)	26.80	60.39	58.02	40.88	61.40	80.70	46.83	*34.68*	82.64
Synthetic WikipediaLarge-500K									
Top-5	**40.85**	94.06	* **66.37**	36.15	57.46	78.73	40.26	28.53	82.97
PS-5	39.22	**101.28**	65.04	37.59	75.74	87.87	47.19	33.84	93.84
Pow-$5_{\beta=0.25}$	38.71	*100.97*	64.39	38.15	75.03	87.52	47.66	34.30	93.52
Pow-$5_{\beta=0.5}$	36.73	99.51	61.36	34.32	77.26	88.63	44.54	31.38	94.08
Log-5	*39.65*	99.69	*65.28*	39.33	68.74	84.37	46.76	33.74	90.67
Macro-R_{prior}-5	33.09	95.08	55.67	29.36	78.33	89.16	39.39	26.86	94.30
Macro-BA_{prior}-5	33.10	95.09	55.68	29.36	78.33	89.16	39.39	26.86	94.30
BCA(Macro-P@5)	18.09	35.29	26.54	**62.69**	29.61	64.80	31.45	22.01	87.59
BCA(Macro-R@5)	30.78	91.07	51.96	26.83	**79.06**	**89.53**	35.80	24.03	*94.80*
BCA(Macro-BA@5)	30.78	91.08	51.96	26.83	**79.06**	**89.53**	35.80	24.03	*94.80*
BCA(Macro-F_1@5)	36.12	93.43	58.71	52.39	70.51	85.25	**57.81**	*44.01*	93.47
BCA(Macro-JS@5)	36.08	93.43	58.59	52.34	70.53	85.26	*57.79*	**44.02**	93.45
BCA(Cov@5)	10.03	40.94	15.85	24.43	61.36	80.68	23.60	14.93	**95.11**
FW(Macro-P@5)	18.08	35.30	26.45	*61.20*	29.12	64.56	30.47	21.40	84.53
FW(Macro-R@5)	30.77	91.13	51.95	26.69	*78.85*	*89.43*	35.61	23.86	94.68
FW(Macro-BA@5)	30.78	91.13	51.96	26.69	*78.85*	*89.43*	35.61	23.86	94.68
FW(Macro-F_1@5)	36.11	93.40	58.69	52.39	70.13	85.06	57.59	43.84	93.19
FW(Macro-JS@5)	36.08	93.41	58.58	52.34	70.15	85.07	57.56	43.85	93.17
Synthetic Amazon-670K									
Top-5	**42.57**	246.20	* **62.77**	41.75	67.00	83.50	48.10	35.83	86.01
PS-5	41.55	**256.18**	61.32	41.29	73.46	86.73	49.51	36.60	91.20
Pow-$5_{\beta=0.25}$	41.99	255.56	61.96	41.99	72.39	86.20	49.87	37.03	90.49
Pow-$5_{\beta=0.5}$	41.25	*255.95*	60.86	40.87	73.75	86.87	49.15	36.22	91.39
Log-5	*42.44*	251.91	*62.59*	42.22	69.76	84.88	49.31	36.74	88.41
Macro-R_{prior}-5	39.97	253.53	58.83	39.42	74.27	87.13	47.61	34.75	91.66
Macro-BA_{prior}-5	39.97	253.53	58.83	39.42	74.27	87.13	47.61	34.75	91.66
BCA(Macro-P@5)	21.46	124.58	29.21	**56.48**	39.63	69.82	40.09	29.22	82.85
BCA(Macro-R@5)	39.23	249.80	57.68	38.64	**75.05**	**87.52**	46.83	33.98	*92.45*
BCA(Macro-BA@5)	39.23	249.80	57.68	38.64	**75.05**	**87.52**	46.83	33.98	*92.45*
BCA(Macro-F_1@5)	39.91	242.67	57.03	50.02	70.14	85.07	**55.56**	*42.38*	91.91
BCA(Macro-JS@5)	39.92	243.11	57.02	49.86	70.20	85.10	*55.48*	**42.39**	91.76
BCA(Cov@5)	26.58	195.18	41.06	39.77	67.92	83.96	42.03	29.75	**93.59**
FW(Macro-P@5)	21.17	124.47	28.85	*54.67*	39.03	69.51	38.84	28.35	80.31
FW(Macro-R@5)	39.25	250.58	57.70	38.98	*74.64*	*87.32*	46.91	34.10	92.09
FW(Macro-BA@5)	39.25	250.58	57.70	38.98	*74.64*	*87.32*	46.91	34.10	92.09
FW(Macro-F_1@5)	39.82	242.70	56.83	49.96	69.33	84.66	54.96	41.95	91.08
FW(Macro-JS@5)	39.83	243.04	56.82	49.83	69.41	84.71	54.93	42.00	90.91

the number of samples in the validation dataset is lower than in the synthetic experiments. Therefore, we use the same training set used to train LPE as a validation set for FW as in Schultheis et al. [2024], since it has been found that this performs better despite the higher risk of overfitting in general.

As before, we add a small constant υ to the denominator of the optimized utilities. While for the BCA algorithm, this value is again υ = 1e-8 in all cases, for FW, we tune this value separately for each optimized utility and dataset to achieve satisfactory results. Increasing the value of υ in this case works as a form of regularization. The larger υ, the smoother the utility function is around 0, reducing the magnitudes of gradients and reducing overfitting at the cost of bias in the objective. The exact values of υ used are provided in Appendix B.1. We select the final υ by splitting the original training set into a training set and an additional validation set. We keep all the other hyperparameters the same, as in the experiments with synthetic labels.

While this regularization significantly improves the results of the FW algorithm, it is still often not enough to help it achieve the best results on datasets with a very small number of samples per label, like Wiki10-31K, WikipediaLarge-500K, and Amazon-670K. On these datasets, the simple tail-labels weighting schemes, like Pow-k, sometimes outperform the classifiers obtained via the FW algorithm.

It is worth noting that by using only the top $k' = 100$ labels, we were able to perform all experiments on a machine with 64 GB of RAM. Although we used different machines and algorithmic variations throughout our research, we do not report specific running times. These range from a few seconds for smaller datasets like EURLex-4K to 1-2 hours of CPU time for larger datasets, such as AmazonCat-14K (in terms of samples) and WikipediaLarge-500K (in terms of labels). All methods are easily parallelizable, allowing to substantially reduce the real running time for larger datasets. Beyond the choice of k', the exact computation time also depends on the optimized metric, the value of k, and the stopping conditions, which can be adjusted to reduce runtime with minimal impact on performance.

9.3 Optimization of mixed utilities

In the previous experiments, we can observe that the optimization of macro-averaged metrics comes at the cost of a significant drop in performance on instance-wise metrics, which may not be desirable. Ideally, we would like to greatly improve performance on tail labels without sacrificing too much of the general performance. To achieve such a trade-off and have good control over it, instead of focusing only on the optimization of tail-label-oriented metrics, we can optimize a utility function that is a convex combination of two or more label-wise utilities, such as general weighted instance-wise utilities, which promote head labels (e.g., precision@k or Hamming score@k), and macro-averaged utilities, which focus on tail-labels performance (e.g., macro-F_β-measure@k). We refer to such a combination of

Table 9.3: Results (%) for $k = 5$ on original XMLC datasets with marginal conditional probabilities coming from PLT model. The green background indicates cells in which the inference algorithm matches the metric it optimizes and the gray text indicates results worse than those results for a given metric. The best results are in **bold**, and the second best are in *italic*. * – while Top-k in general is not optimal for recall@k, we expect it to be the closest to the optimal solution, and we mark it blue background.

Method	Instance @5			Macro @5					
	P	PS	R	P	R	BA	F_1	JS	Cov
RCV1x-2K									
Top-5	**51.67**	*56.92*	* **81.17**	13.37	7.61	53.76	7.62	4.95	35.91
PS-5	*51.23*	**57.21**	*80.69*	19.38	12.18	56.04	12.33	7.83	53.99
Pow-5$_{\beta=0.25}$	50.17	56.63	79.43	18.37	14.16	57.03	13.52	8.54	56.26
Pow-5$_{\beta=0.5}$	47.40	54.47	76.03	16.57	19.80	59.85	15.56	9.87	65.34
Log-5	50.31	56.59	79.50	17.96	12.57	56.23	12.23	7.76	50.95
Macro-R$_{\text{prior}}$-5	34.89	41.94	57.92	14.00	29.11	64.49	14.25	8.75	80.09
Macro-BA$_{\text{prior}}$-5	35.35	42.43	58.73	14.03	29.09	64.48	14.29	8.78	80.09
BCA(Macro-P@5)	28.43	30.36	45.70	**36.36**	3.30	51.57	3.70	2.35	39.20
BCA(Macro-R@5)	29.13	35.64	48.63	13.62	*29.81*	*64.83*	13.18	8.04	*81.15*
BCA(Macro-BA@5)	29.78	36.33	49.82	13.65	**29.82**	**64.84**	13.22	8.07	*81.15*
BCA(Macro-F_1@5)	45.52	51.70	73.47	23.67	17.55	58.72	**18.67**	*11.84*	70.46
BCA(Macro-JS@5)	45.69	51.88	73.69	23.67	17.40	58.64	*18.60*	**11.86**	69.84
BCA(Cov@5)	1.70	4.01	0.96	12.90	22.05	60.92	5.21	2.82	**84.63**
FW(Macro-P@5)	35.32	39.16	57.07	*29.04*	8.77	54.32	11.69	7.24	53.50
FW(Macro-R@5)	34.78	41.82	57.75	14.00	29.11	64.49	14.23	8.73	80.21
FW(Macro-BA@5)	35.26	42.32	58.59	14.04	29.10	64.48	14.28	8.77	80.21
FW(Macro-F_1@5)	46.05	52.22	73.71	23.12	15.51	57.70	17.20	11.09	61.55
FW(Macro-JS@5)	46.30	52.44	74.05	22.77	14.82	57.35	16.60	10.73	60.02
EURLex-4K									
Top-5	57.47	135.71	* 55.66	25.30	22.05	61.00	22.41	17.65	36.71
PS-5	57.36	*144.55*	55.56	27.14	26.73	63.34	25.56	20.20	41.75
Pow-5$_{\beta=0.25}$	*57.48*	142.87	*55.67*	26.60	25.67	62.81	24.85	19.62	40.57
Pow-5$_{\beta=0.5}$	55.40	**145.34**	53.72	26.90	28.47	64.21	26.19	20.60	43.45
Log-5	**57.57**	141.35	**55.75**	26.16	24.76	62.35	24.20	19.11	39.44
Macro-R$_{\text{prior}}$-5	43.99	131.81	42.81	25.27	30.73	65.33	25.19	18.98	46.93
Macro-BA$_{\text{prior}}$-5	44.09	131.97	42.91	25.27	30.74	65.34	25.21	19.00	46.93
BCA(Macro-P@5)	16.88	48.06	16.13	**36.26**	13.93	56.91	16.26	12.72	37.94
BCA(Macro-R@5)	43.29	131.10	42.14	25.07	**31.32**	**65.62**	25.34	19.17	*47.66*
BCA(Macro-BA@5)	43.39	131.24	42.23	25.07	**31.32**	**65.62**	25.35	19.18	*47.66*
BCA(Macro-F_1@5)	50.38	133.52	48.85	*30.46*	27.68	63.81	**27.36**	**21.46**	45.61
BCA(Macro-JS@5)	51.27	134.54	49.63	30.41	27.48	63.71	*27.23*	*21.41*	45.32
BCA(Cov@5)	21.33	83.39	20.57	25.48	27.05	63.48	19.40	13.83	**49.21**
FW(Macro-P@5)	43.39	113.02	42.20	29.62	22.63	61.28	23.01	17.90	40.15
FW(Macro-R@5)	46.78	136.67	45.49	25.14	*30.91*	*65.42*	25.49	19.37	46.56
FW(Macro-BA@5)	46.89	136.83	45.61	25.15	*30.91*	*65.42*	25.50	19.39	46.56
FW(Macro-F_1@5)	52.52	139.16	50.92	27.92	28.05	63.99	26.07	20.34	44.24
FW(Macro-JS@5)	53.82	140.96	52.11	27.09	27.75	63.85	25.69	19.98	43.75
EURLex-4.3K									
Top-5	**68.50**	120.68	* **71.61**	25.36	22.57	61.26	22.73	18.31	34.54
PS-5	*68.40*	*125.92*	*71.55*	27.67	26.79	63.37	26.04	21.19	38.91
Pow-5$_{\beta=0.25}$	68.31	125.22	71.46	26.98	26.13	63.05	25.39	20.64	38.00
Pow-5$_{\beta=0.5}$	66.58	**126.40**	69.83	27.31	28.84	64.40	26.79	21.68	40.62
Log-5	68.37	124.01	71.52	26.25	25.05	62.51	24.54	19.93	36.64
Macro-R$_{\text{prior}}$-5	57.69	117.55	60.74	26.10	31.35	65.65	26.37	20.73	43.74
Macro-BA$_{\text{prior}}$-5	57.84	117.69	60.90	26.10	31.30	65.63	26.37	20.74	43.71
BCA(Macro-P@5)	26.08	47.89	27.05	**36.64**	14.09	57.00	16.48	12.97	38.27
BCA(Macro-R@5)	57.21	116.86	60.26	25.70	**31.66**	*65.80*	26.25	20.70	*43.85*
BCA(Macro-BA@5)	57.42	117.13	60.49	25.72	**31.66**	**65.81**	26.27	20.71	*43.85*
BCA(Macro-F_1@5)	62.38	118.51	65.18	31.25	28.23	64.09	**28.22**	**22.99**	42.51
BCA(Macro-JS@5)	61.70	114.87	64.44	*33.21*	26.02	62.99	*27.87*	*22.89*	41.34
BCA(Cov@5)	15.26	49.12	15.29	26.18	25.11	62.50	18.02	13.22	**45.83**
FW(Macro-P@5)	57.05	104.01	59.83	30.92	22.27	61.11	24.13	19.54	37.25
FW(Macro-R@5)	58.16	118.31	61.21	26.07	*31.43*	65.69	26.42	20.79	43.71
FW(Macro-BA@5)	58.30	118.48	61.36	26.08	31.42	65.69	26.44	20.80	43.69
FW(Macro-F_1@5)	64.13	120.41	67.01	29.46	27.16	63.56	26.85	21.86	40.04
FW(Macro-JS@5)	63.34	117.63	66.11	30.02	25.80	62.88	26.33	21.57	39.12

Method	Instance @5			Macro @5					
	P	PS	R	P	R	BA	F_1	JS	Cov
AmazonCat-13K									
Top-5	**63.96**	81.53	* **74.90**	46.66	32.98	66.48	35.32	27.15	63.81
PS-5	*63.67*	**85.66**	*74.68*	51.29	50.35	75.17	47.74	37.22	79.93
Pow-5$_{\beta=0.25}$	61.78	*84.76*	72.76	49.36	51.36	75.67	47.67	37.05	79.23
Pow-5$_{\beta=0.5}$	56.45	81.66	66.76	41.76	61.10	80.54	46.77	35.18	84.90
Log-5	61.51	83.17	72.20	50.12	45.50	72.74	44.99	35.04	74.83
Macro-R$_{\text{prior}}$-5	39.62	65.19	46.13	27.37	68.86	84.42	34.90	24.33	88.88
Macro-BA$_{\text{prior}}$-5	39.99	65.58	46.75	27.40	68.86	84.42	34.93	24.36	88.88
BCA(Macro-P@5)	31.70	36.00	38.65	**71.42**	14.83	57.40	19.79	14.40	72.59
BCA(Macro-R@5)	35.39	60.42	40.97	27.06	**69.39**	**84.68**	33.97	23.89	*89.35*
BCA(Macro-BA@5)	35.75	60.80	41.60	27.08	**69.39**	**84.68**	33.99	23.91	*89.35*
BCA(Macro-F_1@5)	56.70	77.43	66.88	60.91	49.11	74.54	**52.15**	*41.08*	84.85
BCA(Macro-JS@5)	56.56	77.29	66.65	*60.95*	49.07	74.53	**52.15**	**41.11**	84.79
BCA(Cov@5)	5.96	18.58	5.09	21.13	51.96	75.96	19.19	12.22	**89.82**
FW(Macro-P@5)	48.00	62.80	58.83	58.86	38.32	69.15	40.06	30.21	79.60
FW(Macro-R@5)	39.20	64.71	45.65	27.03	*69.02*	*84.50*	34.51	24.01	89.10
FW(Macro-BA@5)	39.58	65.10	46.28	27.06	*69.02*	*84.50*	34.54	24.04	89.10
FW(Macro-F_1@5)	56.90	78.96	66.95	54.31	53.89	76.93	*51.31*	40.26	83.86
FW(Macro-JS@5)	56.05	78.75	67.09	50.68	55.71	77.85	49.82	38.77	84.75
AmazonCat-14K									
Top-5	**54.63**	63.18	* **83.63**	41.83	36.67	68.33	36.57	27.14	69.59
PS-5	*54.61*	*64.14*	*83.57*	38.93	47.22	73.60	40.47	29.90	79.29
Pow-5$_{\beta=0.25}$	53.57	64.04	82.16	36.10	49.80	74.89	39.75	29.10	79.61
Pow-5$_{\beta=0.5}$	48.58	60.28	75.18	26.96	57.95	78.97	34.54	24.09	84.12
Log-5	54.05	**64.15**	82.61	39.68	45.08	72.53	39.95	29.60	76.51
Macro-R$_{\text{prior}}$-5	31.64	43.57	44.32	17.82	64.21	82.09	24.33	16.16	87.11
Macro-BA$_{\text{prior}}$-5	31.87	43.80	45.17	17.84	64.21	82.09	24.35	16.17	87.11
BCA(Macro-P@5)	25.91	26.45	47.73	**65.03**	8.92	54.45	12.89	8.54	70.53
BCA(Macro-R@5)	29.63	41.21	39.83	17.94	**64.64**	**82.31**	23.87	16.02	*88.18*
BCA(Macro-BA@5)	29.94	41.52	40.58	17.98	**64.64**	**82.31**	23.91	16.06	*88.18*
BCA(Macro-F_1@5)	49.73	59.48	77.24	47.74	43.79	71.89	**43.76**	*32.44*	83.29
BCA(Macro-JS@5)	49.70	59.44	77.20	47.78	43.77	71.88	**43.76**	**32.45**	83.23
BCA(Cov@5)	2.90	7.09	3.19	14.33	41.88	70.92	9.46	5.46	**88.26**
FW(Macro-P@5)	36.75	42.09	59.38	*54.56*	25.06	62.52	30.50	20.94	76.98
FW(Macro-R@5)	31.09	43.06	42.96	17.64	*64.48*	*82.23*	24.04	15.96	87.50
FW(Macro-BA@5)	31.38	43.35	44.09	17.65	*64.48*	*82.23*	24.06	15.98	87.50
FW(Macro-F_1@5)	49.63	57.60	77.08	47.53	43.33	71.66	43.08	32.05	80.22
FW(Macro-JS@5)	49.58	57.59	77.00	47.61	43.28	71.63	*43.09*	32.08	80.10
Wiki10-31K									
Top-5	**63.00**	97.56	* **17.98**	3.73	1.00	50.50	1.42	0.96	4.38
PS-5	*61.83*	128.33	*17.82*	7.91	3.69	51.84	4.51	3.49	9.47
Pow-5$_{\beta=0.25}$	61.06	127.25	17.56	7.40	3.39	51.69	4.19	3.20	9.01
Pow-5$_{\beta=0.5}$	54.03	146.97	15.61	10.03	5.94	52.97	6.77	5.39	12.75
Log-5	56.16	120.69	15.98	6.61	2.91	51.45	3.67	2.72	8.33
Macro-R$_{\text{prior}}$-5	38.04	150.76	10.94	12.04	**8.73**	**54.36**	**9.03**	*7.23*	*16.77*
Macro-BA$_{\text{prior}}$-5	38.28	**150.94**	11.01	12.05	*8.72*	*54.35*	**9.03**	*7.23*	16.75
BCA(Macro-P@5)	34.35	102.57	9.69	**14.44**	6.18	53.08	7.41	6.15	14.62
BCA(Macro-R@5)	27.57	129.69	7.85	12.35	8.48	54.23	*8.98*	7.21	**17.43**
BCA(Macro-BA@5)	27.64	129.78	7.87	12.35	8.48	54.23	*8.98*	7.21	**17.43**
BCA(Macro-F_1@5)	41.01	127.82	11.75	*13.68*	6.67	53.33	8.08	6.58	15.39
BCA(Macro-JS@5)	41.75	128.82	11.96	13.50	6.59	53.29	8.00	6.50	15.24
BCA(Cov@5)	26.95	117.21	7.62	11.15	6.92	53.45	7.46	5.78	16.53
FW(Macro-P@5)	42.77	126.18	12.23	8.46	4.82	52.41	5.61	4.22	12.36
FW(Macro-R@5)	38.02	150.74	10.94	12.04	**8.73**	**54.36**	**9.03**	**7.24**	*16.77*
FW(Macro-BA@5)	38.23	*150.85*	11.00	12.05	**8.73**	**54.36**	**9.03**	**7.24**	16.75
FW(Macro-F_1@5)	31.55	122.28	8.98	11.31	7.74	53.86	7.81	6.30	15.25
FW(Macro-JS@5)	35.10	128.67	10.05	11.62	7.77	53.88	7.98	6.43	15.41

Method	Instance @5			Macro @5					
	P	PS	R	P	R	BA	F_1	JS	Cov
WikiLSHTC-325K									
Top-5	31.14	90.78	* 54.58	18.73	20.72	60.36	17.30	12.97	35.53
PS-5	**32.18**	101.06	**56.72**	19.62	25.45	62.72	19.69	14.64	41.44
Pow-$5_{\beta=0.25}$	*31.43*	98.92	*55.82*	19.35	24.47	62.23	19.18	14.30	40.15
Pow-$5_{\beta=0.5}$	29.97	103.63	54.41	19.12	27.76	63.88	20.28	14.90	44.18
Log-5	31.19	94.56	55.11	19.01	22.36	61.18	18.14	13.58	37.52
Macro-R_{prior}-5	27.35	**105.72**	51.42	17.50	31.21	65.60	20.16	14.33	48.56
Macro-BA_{prior}-5	27.36	**105.72**	51.42	17.50	31.21	65.60	20.16	14.33	48.56
BCA(Macro-P@5)	10.70	34.79	16.98	**33.85**	13.08	56.54	15.37	11.82	36.13
BCA(Macro-R@5)	21.79	96.91	43.59	18.23	**32.81**	**66.40**	20.28	14.43	**52.70**
BCA(Macro-BA@5)	21.79	96.91	43.59	18.23	**32.81**	**66.40**	20.28	14.43	**52.70**
BCA(Macro-F_1@5)	24.87	87.14	44.61	26.86	26.24	63.12	**23.75**	*18.03*	48.59
BCA(Macro-JS@5)	24.29	82.10	43.20	*28.79*	24.34	62.17	*23.69*	**18.15**	46.97
BCA(Cov@5)	15.20	78.05	28.45	19.08	29.12	64.56	17.97	12.72	*50.61*
FW(Macro-P@5)	23.91	78.62	43.47	24.18	22.06	61.03	20.09	15.32	40.17
FW(Macro-R@5)	26.75	*105.58*	50.52	17.61	*31.88*	*65.94*	20.38	14.42	49.60
FW(Macro-BA@5)	26.75	*105.58*	50.52	17.61	*31.88*	*65.94*	20.38	14.42	49.60
FW(Macro-F_1@5)	27.13	92.61	48.82	22.14	25.88	62.94	21.21	15.87	44.13
FW(Macro-JS@5)	27.28	93.04	48.93	21.82	26.03	63.02	21.05	15.69	44.32
WikipediaLarge-500K									
Top-5	37.41	113.72	* 48.07	21.04	21.83	60.92	18.99	14.51	37.87
PS-5	**38.04**	127.08	**49.50**	23.18	27.08	63.54	22.27	16.91	44.82
Pow-$5_{\beta=0.25}$	*37.51*	124.47	*48.93*	22.72	25.88	62.94	21.59	16.42	43.28
Pow-$5_{\beta=0.5}$	35.89	130.97	47.83	23.31	29.38	64.69	23.31	17.55	47.76
Log-5	37.43	118.72	48.47	21.80	23.53	61.77	20.12	15.35	40.18
Macro-R_{prior}-5	31.75	*132.99*	43.96	22.48	33.03	66.51	23.95	17.60	52.56
Macro-BA_{prior}-5	31.75	*132.99*	43.96	22.48	33.03	66.51	23.95	17.60	52.56
BCA(Macro-P@5)	13.66	46.06	15.67	**36.12**	14.28	57.14	16.79	13.11	38.29
BCA(Macro-R@5)	24.92	121.88	36.36	23.33	*34.71*	**67.36**	24.28	17.71	**57.18**
BCA(Macro-BA@5)	24.92	121.88	36.37	23.33	**34.72**	**67.36**	24.28	17.71	**57.18**
BCA(Macro-F_1@5)	29.35	110.76	38.08	29.23	27.87	63.93	**25.62**	*19.60*	51.60
BCA(Macro-JS@5)	28.80	105.36	36.87	*30.82*	26.21	63.10	*25.46*	**19.64**	50.12
BCA(Cov@5)	17.82	98.49	25.15	24.08	30.93	65.46	21.85	15.87	*55.71*
FW(Macro-P@5)	24.71	88.82	32.65	26.36	21.40	60.70	20.20	15.66	40.13
FW(Macro-R@5)	31.02	**133.28**	43.14	22.59	33.91	*66.95*	24.15	17.64	53.83
FW(Macro-BA@5)	31.02	**133.28**	43.14	22.59	33.91	*66.95*	24.15	17.64	53.83
FW(Macro-F_1@5)	30.59	114.01	40.12	25.14	27.22	63.61	22.98	17.43	47.21
FW(Macro-JS@5)	30.87	110.45	39.45	25.12	25.84	62.92	22.19	16.89	44.90
Amazon-670K									
Top-5	36.71	259.09	* 34.39	14.21	14.41	57.20	13.39	11.41	19.79
PS-5	36.71	272.96	34.51	15.24	15.68	57.84	14.50	12.35	21.41
Pow-$5_{\beta=0.25}$	**36.89**	270.81	**34.65**	15.10	15.43	57.71	14.32	12.22	21.07
Pow-$5_{\beta=0.5}$	36.49	**275.49**	34.39	15.58	16.01	58.00	14.81	12.61	21.81
Log-5	*36.84*	264.86	*34.55*	14.61	14.87	57.43	13.81	11.78	20.36
Macro-R_{prior}-5	34.70	*273.87*	32.86	16.02	16.39	*58.19*	15.06	12.76	22.30
Macro-BA_{prior}-5	34.70	*273.87*	32.86	16.02	16.39	*58.19*	15.06	12.76	22.30
BCA(Macro-P@5)	24.08	200.71	23.12	**21.04**	13.67	56.84	15.33	13.52	21.38
BCA(Macro-R@5)	33.28	265.95	31.56	17.62	*16.44*	**58.22**	15.84	*13.57*	**22.98**
BCA(Macro-BA@5)	33.28	265.95	31.56	17.62	*16.44*	**58.22**	15.84	*13.57*	**22.98**
BCA(Macro-F_1@5)	32.25	250.48	30.52	20.00	15.47	57.74	**16.47**	**14.44**	22.61
BCA(Macro-JS@5)	31.91	247.25	30.19	*20.14*	15.27	57.64	*16.40*	**14.44**	22.34
BCA(Cov@5)	29.91	248.56	28.10	16.50	15.63	57.81	14.83	12.49	*22.78*
FW(Macro-P@5)	33.75	256.96	31.85	15.80	15.05	57.52	14.24	12.12	20.73
FW(Macro-R@5)	34.31	273.24	32.53	16.15	**16.45**	**58.22**	15.10	12.79	22.36
FW(Macro-BA@5)	34.31	273.24	32.53	16.15	**16.45**	**58.22**	15.10	12.79	22.36
FW(Macro-F_1@5)	34.91	262.95	32.86	15.60	15.25	57.62	14.32	12.19	20.94
FW(Macro-JS@5)	34.49	259.77	32.47	15.59	15.07	57.54	14.21	12.11	20.73

utilities as mixed utility. Such a combination itself is a label-wise utility (as weighted instance-wise utilities are also label-wise utilities) and can be optimized without any modification by the discussed algorithms. Additionally, a convex combination of functions that are convex and Lipschitz results in a function that preserves these characteristics and, as such, does not violate the guarantees of our algorithms.

As an example, we present the results for mixed utilities that combine instance-wise precision@k with macro-averaged metric:

$$\begin{aligned}\Psi(\widehat{\mathbf{C}}) :=& (1-\lambda)\Psi_{\mathrm{P@}k}(\widehat{\mathbf{C}}) + \lambda\Psi_{\mathrm{macro}}(\widehat{\mathbf{C}}) \\ =& \sum_{j=1}^{m}(1-\lambda)\psi_{\mathrm{P@}k}(\widehat{\boldsymbol{c}}_j) + \lambda\psi_{\mathrm{macro}}(\widehat{\boldsymbol{c}}_j)\end{aligned} \tag{9.1}$$

In Figures 9.1 and 9.2, we present the results of optimization of such mixed utility of precision@k with macro-recall@k, macro-F_β-measure@k, and coverage@k with different values of $\lambda \in \{0, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 0.95, 1\}$, where $\lambda = 0$ is equivalent of Top-k. As in the case of previous results, here we present the results for both synthetic and original datasets for $k = 5$, and in Figures B.1 and B.2 for $k \in \{1, 3\}$. The presented results are the mean values over 5 label generations in the case of synthetic datasets and 5 runs in the case of original datasets. The plots show that the instance-vs-macro curve has a nice concave shape. In synthetic experiments, the curve always dominates simple baselines. In particular, we can initially improve macro-averaged metrics with only a minor drop in instance-wise performance. Only when we want to optimize even more strongly for tail labels, we will get larger drops on head labels. Because of the plug-in nature of BCA and FW methods working on top of LPE, it is relatively easy and cheap to tune λ, so one can easily select an optimal interpolation constant according to some criteria, such as a maximum decrease of instance-wise performance.

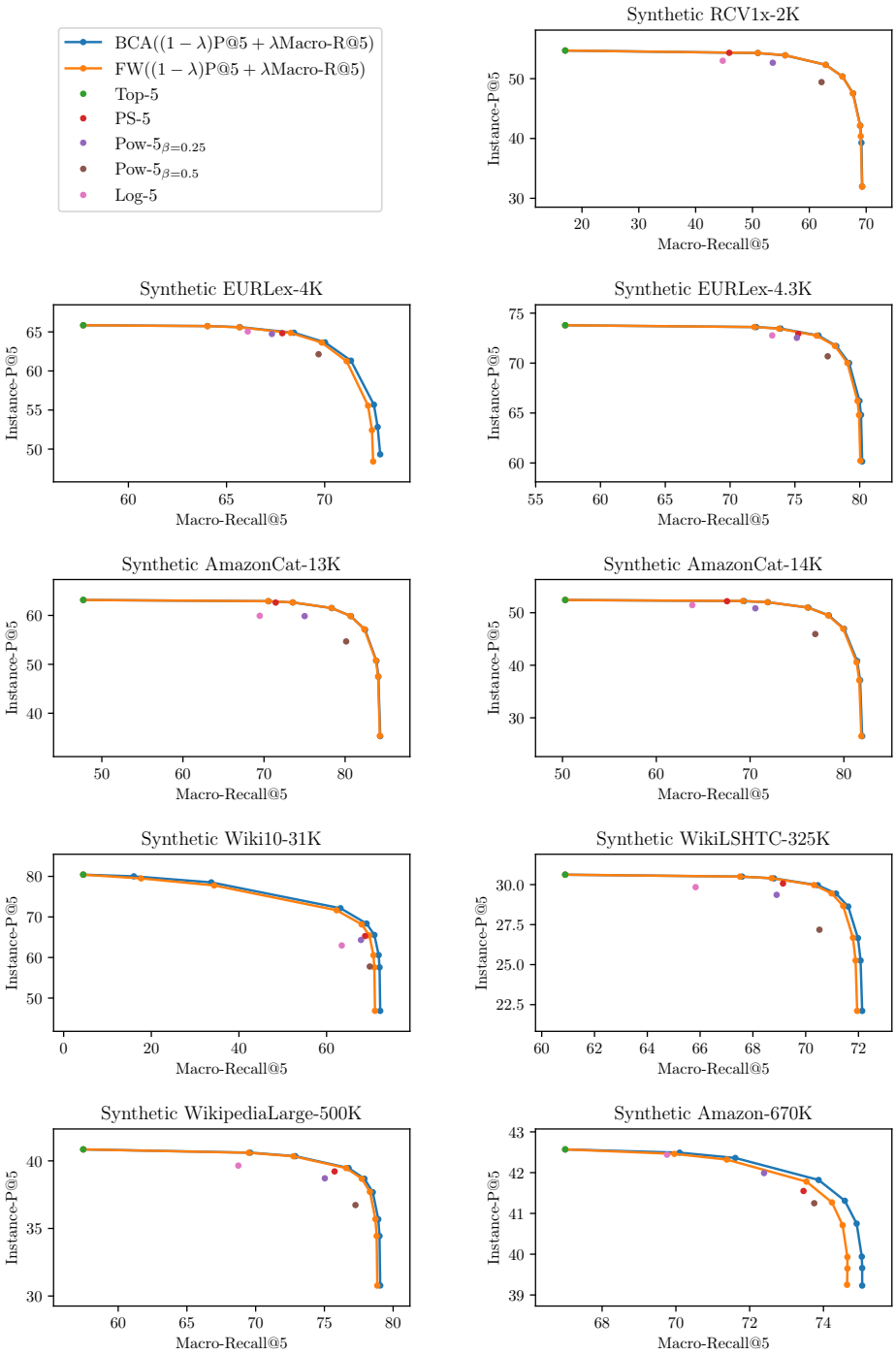
9.4 Efficiency of PLT with BF*-search

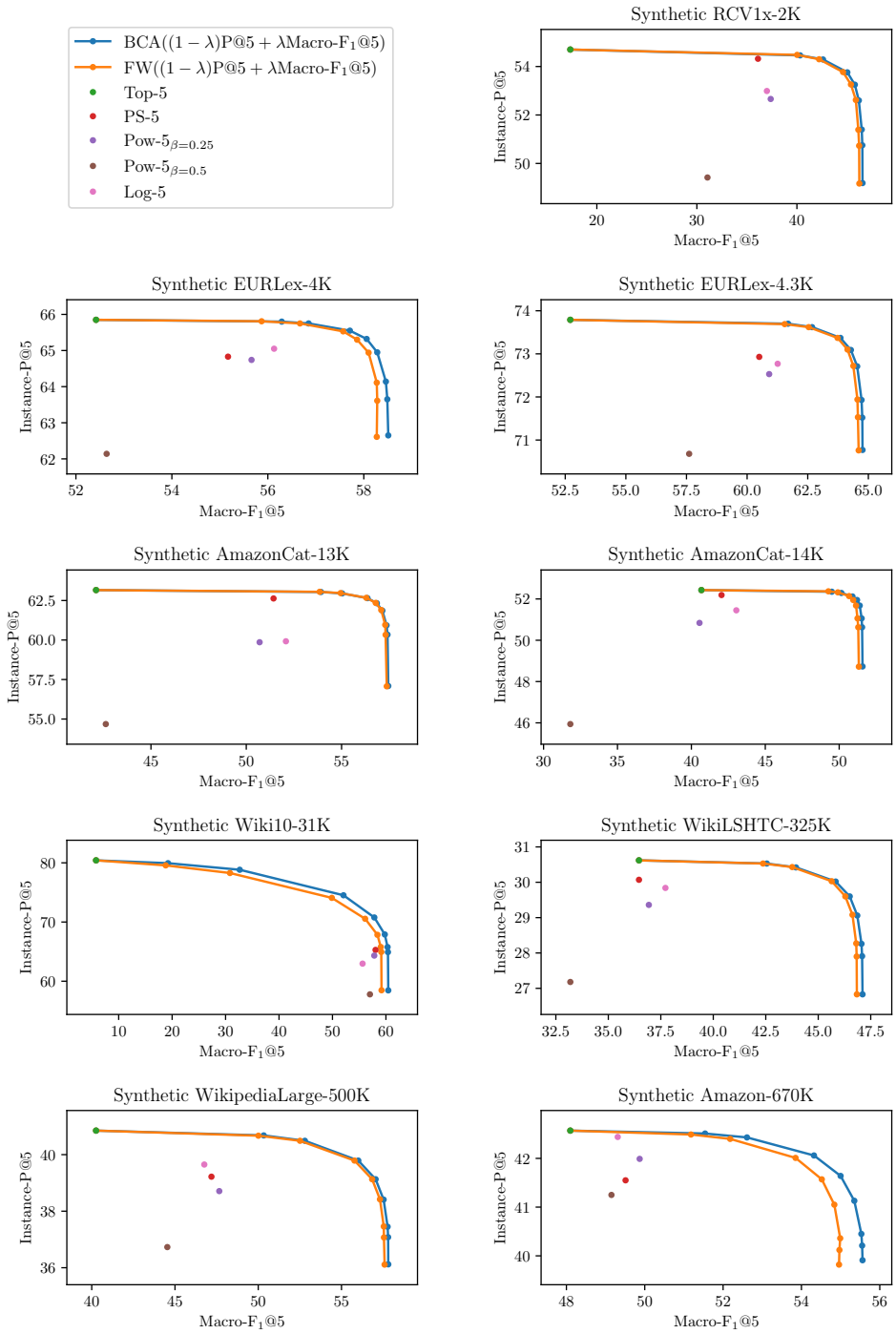
Finally, in this last experiment, we compare the computational performance of two approaches to efficient inference:

- a general two-step approach that first predicts k' top conditional marginal probabilities $\widehat{\boldsymbol{\eta}}(\boldsymbol{x})$ and then applies reweighting according to one of the methods such as PS-k, Pow-k, Macro-$\mathrm{R}_{\mathrm{prior}}$-$k$, or a random classifier obtained using FW($\cdot$),
- direct inference using BF*-search in PLT for finding k labels with highest values of $a\widehat{\eta}(\boldsymbol{x}) + b$ (as described in Section 8.2.3).

We are interested in investigating if BF*-search for PLT may bring even further speed up over the two-step approach. Because of that, we compare inference times of BF*-search for different objectives and budgets $k = \{1, 3, 5\}$, with a time of

Figure 9.1: Results (%) for $k = 5$ of optimization of mixed utilities on synthetic versions of XMLC datasets with ideal estimates of marginal conditional probabilities $\boldsymbol{\eta}(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}(\boldsymbol{x})$.




BCA($(1-\lambda)$P@5 + λMacro-F$_1$@5)
FW($(1-\lambda)$P@5 + λMacro-F$_1$@5)
Top-5
PS-5
Pow-5$_{\beta=0.25}$
Pow-5$_{\beta=0.5}$
Log-5
Synthetic RCV1x-2K
Synthetic EURLex-4K
Synthetic EURLex-4.3K
Synthetic AmazonCat-13K
Synthetic AmazonCat-14K
Synthetic Wiki10-31K
Synthetic WikiLSHTC-325K
Synthetic WikipediaLarge-500K
Synthetic Amazon-670K
Instance-P@5
Macro-F$_1$@5

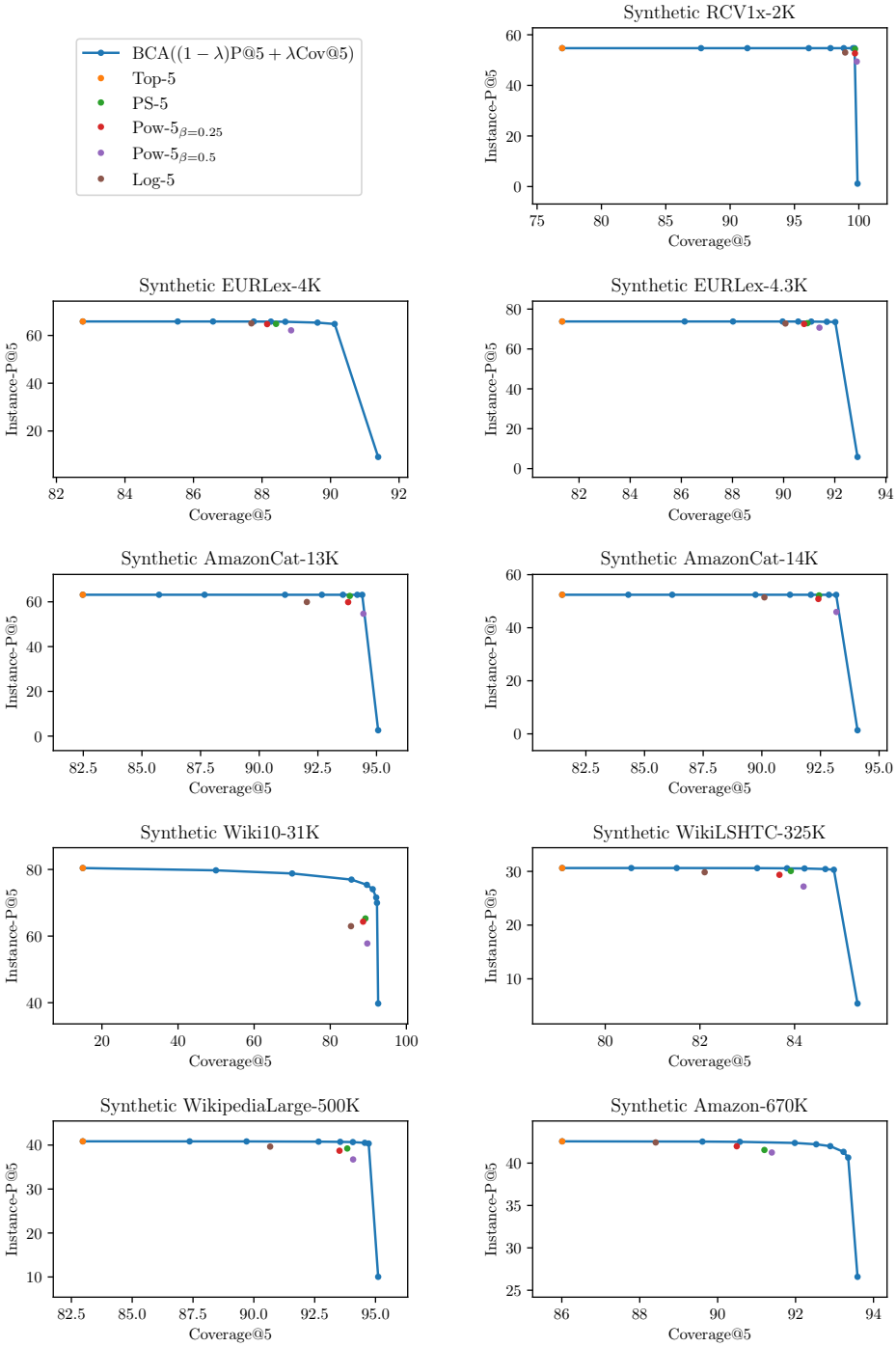
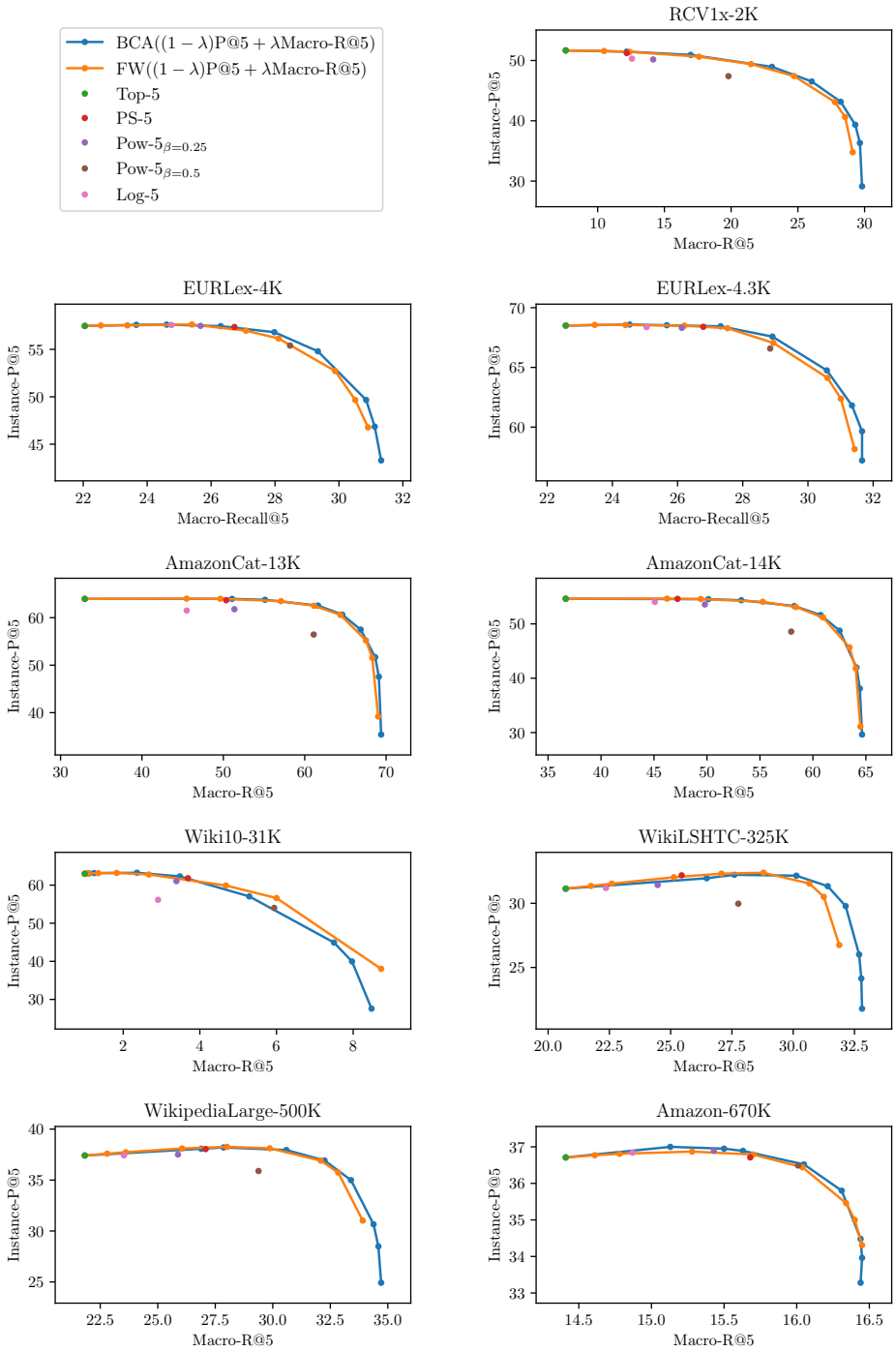
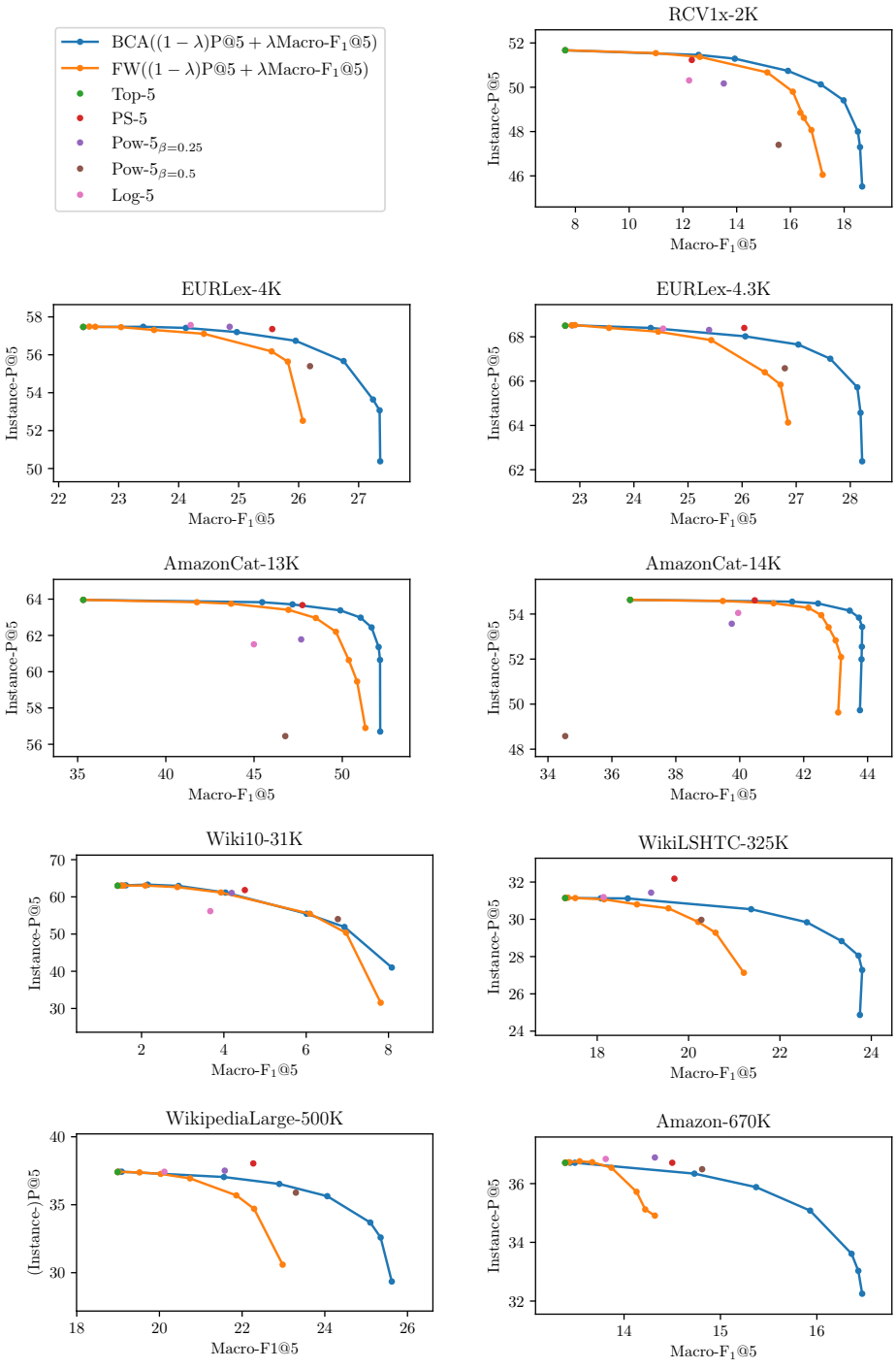
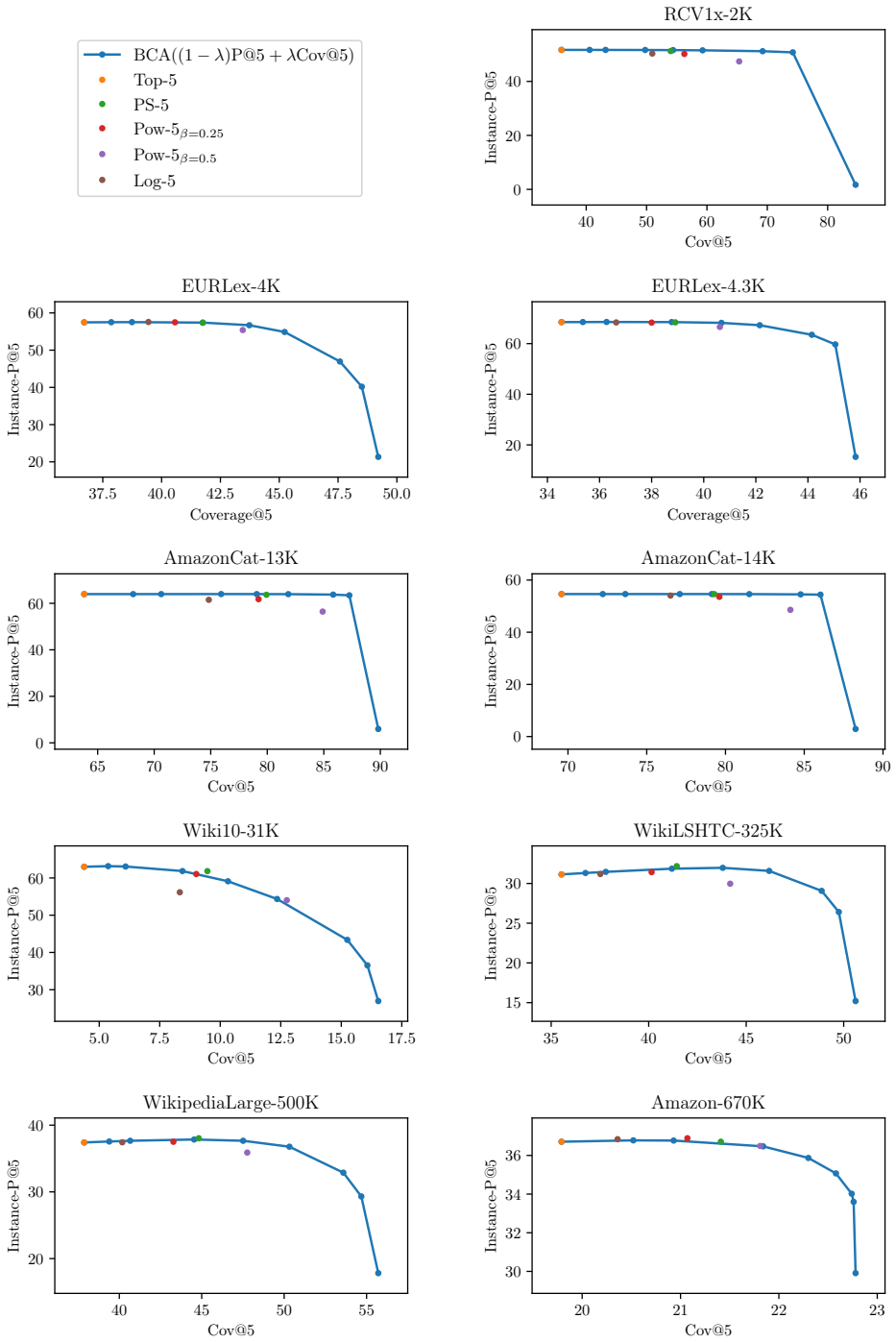

BCA((1 − λ)P@5 + λCov@5)
Top-5
PS-5
Pow-5β=0.25
Pow-5β=0.5
Log-5
Synthetic RCV1x-2K
Synthetic EURLex-4K
Synthetic EURLex-4.3K
Synthetic AmazonCat-13K
Synthetic AmazonCat-14K
Synthetic Wiki10-31K
Synthetic WikiLSHTC-325K
Synthetic WikipediaLarge-500K
Synthetic Amazon-670K
Instance-P@5
Coverage@5

Figure 9.2: Results (%) for $k = 5$ of optimization of mixed utilities on original XMLC datasets with marginal conditional probabilities coming from the PLT model.




BCA((1 − λ)P@5 + λMacro-F1@5)
FW((1 − λ)P@5 + λMacro-F1@5)
Top-5
PS-5
Pow-5β=0.25
Pow-5β=0.5
Log-5
RCV1x-2K
EURLex-4K
EURLex-4.3K
AmazonCat-13K
AmazonCat-14K
Wiki10-31K
WikiLSHTC-325K
WikipediaLarge-500K
Amazon-670K
Instance-P@5
(Instance-)P@5
Instance-P@5
Macro-F1@5
Macro-F1@5
Macro-F1@5


BCA((1 − λ)P@5 + λCov@5)
Top-5
PS-5
Pow-5β=0.25
Pow-5β=0.5
Log-5
RCV1x-2K
EURLex-4K
EURLex-4.3K
AmazonCat-13K
AmazonCat-14K
Wiki10-31K
WikiLSHTC-325K
WikipediaLarge-500K
Amazon-670K
Instance-P@5
Coverage@5
Cov@5

obtaining marginals for top $k' = 100$ labels, using the same PLT model as in previous experiments. The $k' = 100$ labels need to be later reweighted and the final top k labels selected. Because the speed of these operations may vary slightly depending on $\boldsymbol{a}$ and $\boldsymbol{b}$, we only report the time of obtaining $k' = 100$ labels. We present the results in Table 9.4. We do not report results for utilities themselves, as they are almost identical, as BF*-search gives only a very tiny improvement to reweighting top $k' = 100$ labels. A similar result was obtained by [Schultheis et al., 2023], where a two-step approach with $k' = 100$ and $k' = 1000$ was compared, and $k' = 1000$ was only slightly better than $k' = 100$.

We observe that BF*-search is an even faster inference method in the case of almost all objectives, with the exception of BF*(Macro-R$_{\text{prior}}$-k), BF*(FW(Macro-R@k)), and BF*(FW(Macro-F$_1$@k)). Those have the largest differences between values in $\boldsymbol{a}$ and $\boldsymbol{b}$, forcing BF*-search to explore nodes with tiny probabilities in order to find the exact solution. It is worth noting that the tree structure and the type of node estimator have an impact on the performance of PLT-based algorithms [Jasinska-Kobus et al., 2020]. Although here we limited ourselves to just a single setup of tree structures, a popular binary tree on top of a large cluster of labels (of size 400), we showed that PLT-based inference can be an alternative to the two-step inference with reweighting k' labels with the highest $\eta_j(\boldsymbol{x})$.

9.5 Summary of the chapter

In this chapter, we conducted extensive empirical experiments on a wide range of benchmark datasets from the well-known XMLC repository, considering instance-wise and macro-averaged utility functions. We first conducted experiments with synthetic labels, designed to compare inference algorithms, emulating perfect knowledge of label conditional marginal probabilities and, as such, eliminating the main source of regret of all the methods. The results show that all the methods are the best on their targeted utilities, confirming our theoretical results. We then conduct experiments using original labels to reflect a more realistic scenario of imperfect probability estimates and high data sparsity. Here, BCA methods increased their advantage. The gap between BCA and FW became larger due to challenges like overfitting, which is especially notable for macro-precision. Regularization helped mitigate this issue, though simpler inference strategies (Top-k) also showed robust performance in some cases. Additionally, we demonstrated that BCA and FW allow for the optimization of mixed utilities that can lead to a practical approach of balancing the head- and tail-label performance. Finally, we investigated the computational efficiency of the PLT inference based on the BF*-search, showing that it can be computationally less expensive compared to the simpler two-step strategy in which prediction of marginals is followed by reweighting.

Table 9.4: Comparison of average inference times per instance (ms) of predicting the top 100 labels first and doing reweighting (Top-100) with BF$^\star(\cdot)$ algorithm applied to predicting with different weights and $k \in \{1, 3, 5\}$. The time is reported in milliseconds (ms) and the speed-up is reported as a ratio of the time taken by BF$^\star(\cdot)$ algorithm to the time of Top-100.

Method	$k=1$		$k=3$		$k=5$	
	T/n_{test}	(speed-up)	T/n_{test}	(speed-up)	T/n_{test}	(speed-up)
RCV1x-2K						
Top-100	11.20	(1.00x)	11.20	(1.00x)	11.20	(1.00x)
Top-k	1.17	(9.60x)	1.74	(6.46x)	2.34	(4.79x)
BF*(PS-k)	2.36	(4.76x)	3.15	(3.55x)	4.12	(2.72x)
BF*(Pow-$k_{\beta=0.25}$)	2.59	(4.33x)	3.40	(3.30x)	4.43	(2.53x)
BF*(Pow-$k_{\beta=0.5}$)	4.13	(2.71x)	5.42	(2.07x)	6.73	(1.66x)
BF*(Log-k)	2.03	(5.52x)	2.69	(4.17x)	3.46	(3.24x)
BF*(Macro-R$_{\text{prior}}$-k)	9.00	(1.24x)	10.21	(1.10x)	11.28	(0.99x)
BF*(FW(0.9P@k+0.1Macro-R@k))	3.46	(3.24x)	5.99	(1.87x)	8.57	(1.31x)
BF*(FW(0.7P@k+0.3Macro-R@k))	4.84	(2.32x)	7.68	(1.46x)	10.20	(1.10x)
BF*(FW(Macro-R@k))	10.69	(1.05x)	11.97	(0.94x)	12.25	(0.91x)
BF*(FW(0.9P@k+0.1Macro-F$_1$@k))	3.42	(3.27x)	4.77	(2.35x)	7.12	(1.57x)
BF*(FW(0.7P@k+0.3Macro-F$_1$@k))	4.54	(2.47x)	6.24	(1.80x)	8.87	(1.26x)
BF*(FW(Macro-F$_1$@k))	10.87	(1.03x)	11.98	(0.94x)	12.21	(0.92x)
EURLex-4K						
Top-100	16.37	(1.00x)	16.37	(1.00x)	16.37	(1.00x)
Top-k	2.69	(6.08x)	4.53	(3.62x)	5.16	(3.17x)
BF*(PS-k)	5.04	(3.25x)	6.32	(2.59x)	7.41	(2.21x)
BF*(Pow-$k_{\beta=0.25}$)	4.52	(3.62x)	5.65	(2.90x)	6.87	(2.38x)
BF*(Pow-$k_{\beta=0.5}$)	6.59	(2.48x)	8.14	(2.01x)	9.34	(1.75x)
BF*(Log-k)	3.61	(4.53x)	4.69	(3.49x)	5.79	(2.83x)
BF*(Macro-R$_{\text{prior}}$-k)	20.40	(0.80x)	22.94	(0.71x)	23.01	(0.71x)
BF*(FW(0.9P@k+0.1Macro-R@k))	6.84	(2.39x)	8.33	(1.96x)	10.77	(1.52x)
BF*(FW(0.7P@k+0.3Macro-R@k))	7.68	(2.13x)	10.71	(1.53x)	13.29	(1.23x)
BF*(FW(Macro-R@k))	16.46	(0.99x)	19.86	(0.82x)	19.41	(0.84x)
BF*(FW(0.9P@k+0.1Macro-F$_1$@k))	6.62	(2.47x)	6.54	(2.50x)	8.19	(2.00x)
BF*(FW(0.7P@k+0.3Macro-F$_1$@k))	8.02	(2.04x)	8.22	(1.99x)	10.53	(1.55x)
BF*(FW(Macro-F$_1$@k))	15.10	(1.08x)	18.12	(0.90x)	19.52	(0.84x)
EURLex-4.3K						
Top-100	106.16	(1.00x)	106.16	(1.00x)	106.16	(1.00x)
Top-k	20.71	(5.12x)	30.44	(3.49x)	38.14	(2.78x)
BF*(PS-k)	44.58	(2.38x)	47.77	(2.22x)	56.96	(1.86x)
BF*(Pow-$k_{\beta=0.25}$)	37.60	(2.82x)	45.72	(2.32x)	53.82	(1.97x)
BF*(Pow-$k_{\beta=0.5}$)	51.96	(2.04x)	61.79	(1.72x)	72.40	(1.47x)
BF*(Log-k)	32.53	(3.26x)	38.70	(2.74x)	46.39	(2.29x)
BF*(Macro-R$_{\text{prior}}$-k)	91.31	(1.16x)	103.38	(1.03x)	114.13	(0.93x)
BF*(FW(0.9P@k+0.1Macro-R@k))	45.08	(2.35x)	63.18	(1.68x)	79.06	(1.34x)
BF*(FW(0.7P@k+0.3Macro-R@k))	57.04	(1.86x)	77.75	(1.37x)	96.17	(1.10x)
BF*(FW(Macro-R@k))	101.64	(1.04x)	117.02	(0.91x)	124.01	(0.86x)
BF*(FW(0.9P@k+0.1Macro-F$_1$@k))	47.34	(2.24x)	53.70	(1.98x)	70.67	(1.50x)
BF*(FW(0.7P@k+0.3Macro-F$_1$@k))	55.07	(1.93x)	67.08	(1.58x)	84.09	(1.26x)
BF*(FW(Macro-F$_1$@k))	102.47	(1.04x)	116.74	(0.91x)	123.05	(0.86x)

Method	$k=1$		$k=3$		$k=5$	
	T/n_{test}	(speed-up)	T/n_{test}	(speed-up)	T/n_{test}	(speed-up)
AmazonCat-13K						
Top-100	19.82	(1.00x)	19.82	(1.00x)	19.82	(1.00x)
Top-k	1.16	(17.10x)	1.94	(10.20x)	2.60	(7.62x)
BF*(PS-k)	3.63	(5.46x)	4.51	(4.39x)	5.95	(3.33x)
BF*(Pow-$k_{\beta=0.25}$)	3.77	(5.26x)	4.68	(4.24x)	6.14	(3.23x)
BF*(Pow-$k_{\beta=0.5}$)	7.04	(2.82x)	9.30	(2.13x)	12.07	(1.64x)
BF*(Log-k)	2.71	(7.31x)	3.22	(6.15x)	4.06	(4.88x)
BF*(Macro-R$_{\text{prior}}$-k)	19.22	(1.03x)	26.11	(0.76x)	30.12	(0.66x)
BF*(FW(0.9P@k+0.1Macro-R@k))	8.10	(2.45x)	13.70	(1.45x)	19.57	(1.01x)
BF*(FW(0.7P@k+0.3Macro-R@k))	12.09	(1.64x)	19.45	(1.02x)	25.01	(0.79x)
BF*(FW(Macro-R@k))	27.48	(0.72x)	32.95	(0.60x)	36.01	(0.55x)
BF*(FW(0.9P@k+0.1Macro-F$_1$@k))	5.93	(3.34x)	8.00	(2.48x)	13.18	(1.50x)
BF*(FW(0.7P@k+0.3Macro-F$_1$@k))	7.84	(2.53x)	11.28	(1.76x)	17.96	(1.10x)
BF*(FW(Macro-F$_1$@k))	27.22	(0.73x)	33.38	(0.59x)	36.69	(0.54x)
AmazonCat-14K						
Top-100	23.77	(1.00x)	23.77	(1.00x)	23.77	(1.00x)
Top-k	0.56	(42.40x)	2.32	(10.24x)	3.88	(6.13x)
BF*(PS-k)	0.90	(26.53x)	8.94	(2.66x)	11.24	(2.11x)
BF*(Pow-$k_{\beta=0.25}$)	1.29	(18.43x)	8.76	(2.71x)	10.58	(2.25x)
BF*(Pow-$k_{\beta=0.5}$)	6.79	(3.50x)	17.13	(1.39x)	20.30	(1.17x)
BF*(Log-k)	0.78	(30.43x)	5.00	(4.75x)	6.24	(3.81x)
BF*(Macro-R$_{\text{prior}}$-k)	31.95	(0.74x)	35.34	(0.67x)	39.03	(0.61x)
BF*(FW(0.9P@k+0.1Macro-R@k))	4.23	(5.62x)	28.14	(0.84x)	33.21	(0.72x)
BF*(FW(0.7P@k+0.3Macro-R@k))	7.89	(3.01x)	32.79	(0.72x)	36.14	(0.66x)
BF*(FW(Macro-R@k))	32.88	(0.72x)	36.73	(0.65x)	37.79	(0.63x)
BF*(FW(0.9P@k+0.1Macro-F$_1$@k))	3.34	(7.11x)	18.52	(1.28x)	25.40	(0.94x)
BF*(FW(0.7P@k+0.3Macro-F$_1$@k))	3.47	(6.85x)	24.24	(0.98x)	30.79	(0.77x)
BF*(FW(Macro-F$_1$@k))	32.75	(0.73x)	36.44	(0.65x)	38.08	(0.62x)
Wiki10-31K						
Top-100	336.66	(1.00x)	336.66	(1.00x)	336.66	(1.00x)
Top-k	9.14	(36.85x)	30.04	(11.21x)	47.47	(7.09x)
BF*(PS-k)	105.95	(3.18x)	131.25	(2.56x)	156.01	(2.16x)
BF*(Pow-$k_{\beta=0.25}$)	100.18	(3.36x)	124.07	(2.71x)	148.39	(2.27x)
BF*(Pow-$k_{\beta=0.5}$)	202.67	(1.66x)	248.39	(1.36x)	274.51	(1.23x)
BF*(Log-k)	72.43	(4.65x)	101.88	(3.30x)	120.67	(2.79x)
BF*(Macro-R$_{\text{prior}}$-k)	316.70	(1.06x)	354.81	(0.95x)	377.35	(0.89x)
BF*(FW(0.9P@k+0.1Macro-R@k))	50.82	(6.62x)	100.79	(3.34x)	147.92	(2.28x)
BF*(FW(0.7P@k+0.3Macro-R@k))	90.64	(3.71x)	175.54	(1.92x)	236.92	(1.42x)
BF*(FW(Macro-R@k))	382.30	(0.88x)	409.93	(0.82x)	415.45	(0.81x)
BF*(FW(0.9P@k+0.1Macro-F$_1$@k))	68.72	(4.90x)	132.47	(2.54x)	134.18	(2.51x)
BF*(FW(0.7P@k+0.3Macro-F$_1$@k))	121.92	(2.76x)	121.44	(2.77x)	143.72	(2.34x)
BF*(FW(Macro-F$_1$@k))	379.41	(0.89x)	408.91	(0.82x)	416.81	(0.81x)

Method	$k=1$		$k=3$		$k=5$	
	T/n_{test}	(speed-up)	T/n_{test}	(speed-up)	T/n_{test}	(speed-up)
WikiLSHTC-325K						
Top-100	99.92	(1.00x)	99.92	(1.00x)	99.92	(1.00x)
Top-k	5.69	(17.57x)	12.27	(8.14x)	18.01	(5.55x)
BF*(PS-k)	19.54	(5.11x)	35.97	(2.78x)	49.37	(2.02x)
BF*(Pow-$k_{\beta=0.25}$)	15.58	(6.41x)	28.46	(3.51x)	38.03	(2.63x)
BF*(Pow-$k_{\beta=0.5}$)	32.12	(3.11x)	55.78	(1.79x)	71.97	(1.39x)
BF*(Log-k)	8.75	(11.42x)	17.20	(5.81x)	23.94	(4.17x)
BF*(Macro-R$_{\text{prior}}$-k)	97.52	(1.02x)	141.89	(0.70x)	167.93	(0.59x)
BF*(FW(0.9P@k+0.1Macro-R@k))	34.38	(2.91x)	93.03	(1.07x)	139.76	(0.71x)
BF*(FW(0.7P@k+0.3Macro-R@k))	59.97	(1.67x)	138.18	(0.72x)	183.10	(0.55x)
BF*(FW(Macro-R@k))	152.93	(0.65x)	206.38	(0.48x)	225.94	(0.44x)
BF*(FW(0.9P@k+0.1Macro-F$_1$@k))	43.04	(2.32x)	45.69	(2.19x)	71.30	(1.40x)
BF*(FW(0.7P@k+0.3Macro-F$_1$@k))	70.59	(1.42x)	71.64	(1.39x)	110.02	(0.91x)
BF*(FW(Macro-F$_1$@k))	153.01	(0.65x)	207.76	(0.48x)	231.45	(0.43x)
WikipediaLarge-500K						
Top-100	459.20	(1.00x)	459.20	(1.00x)	459.20	(1.00x)
Top-k	26.15	(17.56x)	56.33	(8.15x)	79.56	(5.77x)
BF*(PS-k)	75.81	(6.06x)	136.62	(3.36x)	183.59	(2.50x)
BF*(Pow-$k_{\beta=0.25}$)	64.49	(7.12x)	111.51	(4.12x)	148.99	(3.08x)
BF*(Pow-$k_{\beta=0.5}$)	116.60	(3.94x)	200.64	(2.29x)	265.88	(1.73x)
BF*(Log-k)	39.59	(11.60x)	73.06	(6.29x)	98.65	(4.65x)
BF*(Macro-R$_{\text{prior}}$-k)	294.13	(1.56x)	488.64	(0.94x)	596.96	(0.77x)
BF*(FW(0.9P@k+0.1Macro-R@k))	77.79	(5.90x)	225.33	(2.04x)	367.11	(1.25x)
BF*(FW(0.7P@k+0.3Macro-R@k))	138.73	(3.31x)	370.01	(1.24x)	551.06	(0.83x)
BF*(FW(Macro-R@k))	480.54	(0.96x)	732.23	(0.63x)	887.14	(0.52x)
BF*(FW(0.9P@k+0.1Macro-F$_1$@k))	89.17	(5.15x)	162.44	(2.83x)	242.30	(1.90x)
BF*(FW(0.7P@k+0.3Macro-F$_1$@k))	117.32	(3.91x)	235.76	(1.95x)	380.11	(1.21x)
BF*(FW(Macro-F$_1$@k))	476.89	(0.96x)	718.68	(0.64x)	879.73	(0.52x)
Amazon-670K						
Top-100	143.00	(1.00x)	143.00	(1.00x)	143.00	(1.00x)
Top-k	16.60	(8.62x)	25.95	(5.51x)	32.92	(4.34x)
BF*(PS-k)	32.70	(4.37x)	47.13	(3.03x)	56.14	(2.55x)
BF*(Pow-$k_{\beta=0.25}$)	28.24	(5.06x)	40.62	(3.52x)	49.24	(2.90x)
BF*(Pow-$k_{\beta=0.5}$)	34.20	(4.18x)	48.28	(2.96x)	58.84	(2.43x)
BF*(Log-k)	20.44	(7.00x)	31.16	(4.59x)	38.15	(3.75x)
BF*(Macro-R$_{\text{prior}}$-k)	71.44	(2.00x)	94.41	(1.51x)	112.15	(1.28x)
BF*(FW(0.9P@k+0.1Macro-R@k))	25.97	(5.51x)	52.61	(2.72x)	73.95	(1.93x)
BF*(FW(0.7P@k+0.3Macro-R@k))	39.65	(3.61x)	78.99	(1.81x)	105.76	(1.35x)
BF*(FW(Macro-R@k))	95.02	(1.50x)	123.60	(1.16x)	140.87	(1.02x)
BF*(FW(0.9P@k+0.1Macro-F$_1$@k))	28.48	(5.02x)	43.16	(3.31x)	54.73	(2.61x)
BF*(FW(0.7P@k+0.3Macro-F$_1$@k))	38.87	(3.68x)	58.97	(2.42x)	77.91	(1.84x)
BF*(FW(Macro-F$_1$@k))	95.66	(1.49x)	124.86	(1.15x)	145.72	(0.98x)

10 Summary

In this thesis, we summarized, unified under a single notation, and extended the results we published in the area of extreme multi-label classification (XMLC) regarding the problem of optimal inference for different types of utilities under prediction budgeted at k (the classifier needs to predict exactly k labels for each instance), which is a popular setting in many applications of XMLC like recommender systems or online advertising. All inference algorithms considered in this dissertation work on top of the Label Probability Estimator (LPE), which estimates the marginal conditional probabilities of labels $\eta_j(\boldsymbol{x}) = \mathbb{P}[y_j = 1 \,|\, \boldsymbol{x}]$, making our work agnostic to the specific LPE type and broadly applicable to a wide range of methods used in XMLC.

We started our analysis with popular instance-wise utilities and took a closer look at popular precision@k and similar metrics such as the Hamming score@k. We considered their weighted variants and generalized them under the family of instance-wise weighted utilities. For these utilities, we demonstrated the form of the Bayes (optimal) classifier, which is determined by $\eta_j(\boldsymbol{x})$, and derived an upper bound for the regret that one suffers when using inaccurate estimates of $\eta_j(\boldsymbol{x})$ instead. We also applied a similar analysis for popular metrics such as recall@k and (n)DCG@k.

Then, we took a close look at these metrics under the missing labels setting. We demonstrated that under the propensity model $p_j(\boldsymbol{x}) \coloneqq \mathbb{P}\left[\breve{y}_j = 1 \,|\, y_j = 1, \boldsymbol{x}\right]$, the unbiased estimates of instance-wise weighted utilities remain in the class of instance-wise weighted utilities. We then discussed the popular empirical propensity model proposed by Jain et al. [2016], which became a standard approach to evaluating long-tailed performance via propensity-scored metrics, especially propensity-scored precision@k. We highlighted problems with evaluating performance on rare labels using this approach and advocated for the development of metrics better suited for addressing tail labels.

As a promising alternative, we turned to a family of label-wise utilities, especially macro-averaged matrices, that first calculate a binary metric on each label separately and then average the results, treating performance for every label

equally, despite their frequency. While optimization of macro-averaged metrics is a well-researched problem and boils down to finding the optimal threshold on $\eta_j(\boldsymbol{x})$ for each label separately, it turns into a much harder problem under the budget "at k" constraint. As label-wise utilities belong to the family of general metrics defined on the confusion matrix, we analyzed the problem of finding the Bayes classifier for label-wise metrics with a budget "at k" constraint under the expected test utility (ETU) framework, where we aim to maximize the expected value of a metric on a given test set, and the population utility (PU) framework, where the goal is to maximize the metric on the expected confusion matrix on a population level.

Under the ETU framework, we showed that the form of the optimal classifier is defined on $\eta_j(\boldsymbol{x})$. However, calculating it is computationally intractable with a large number of instances and labels. Because of that, we introduce a linear approximation of the ETU objective and propose a block coordinate ascent (BCA) algorithm to maximize it. We also demonstrated regret guarantees depending on the properties of optimized utility and accuracy of provided $\eta_j(\boldsymbol{x})$ estimates.

Analogously, under the PU framework, we showed that the optimal classifier belongs to the class of randomized classifiers. Additionally, when certain conditions of $\boldsymbol{\eta}(\boldsymbol{x})$ distribution and utility are met, the optimal classifier is a linear function of a confusion matrix. Next, we have introduced a consistent Frank-Wolfe (FW) algorithm, which is capable of finding an optimal randomized classifier by performing optimization over the set of feasible confusion matrices. Once again, we show regret guarantees depending on the properties of optimized utility and error of $\eta_j(\boldsymbol{x})$ estimates.

The computational complexity of all introduced inference algorithms is linear with the number of labels, which is problematic in the context of XMLC. For this reason, as a last algorithmic contribution to this work, we focus on reducing the computational complexity of the introduced algorithms. First, we propose to leverage the sparsity of labels and consider only k' labels with the highest marginal probabilities, since most of $\eta_j(\boldsymbol{x})$ is equal or very close to zero. Many XMLC methods support the retrieval of a subset of labels with the highest probabilities or scores, making this method generally applicable. As an alternative approach, we propose to use probabilistic label trees (PLT), which organize labels into a tree structure and decompose their $\eta_j(\boldsymbol{x})$ over a path from the root to the label node. This allows for the application of classical search algorithms like uniform-cost search or beam-search to efficiently find the labels with the highest $\eta_j(\boldsymbol{x})$. In this work, we introduced a more general tree search algorithm based on a best-first-star search that can efficiently find labels with the highest value of $a_j\eta_j(\boldsymbol{x}) + b_j$ that may improve prediction speed even further.

Finally, to conclude this thesis, we performed a comprehensive set of experiments using both synthetic and real-world data sets, confirming our theoretical findings and showcasing the practicality of the introduced algorithms. We also demonstrated that under our framework, one can optimize a convex combination of instance-wise metrics with label-wise objectives to control the trade-off of performance between them effectively. We find that it is possible to significantly

improve the quality of tail-label predictions with minimal compromise to head-label performance.

Ultimately, we believe that this thesis will contribute to the shift toward using macro-averaged metrics for the evaluation of long-tail performance in XMLC, especially under the novel setting of the budget "at k" to which understanding, both theoretically and practically, we significantly contributed in this work. We hope that the introduced methods, thanks to their general applicability and independence from specific label probability estimators, will stand the test of time, remaining relevant in ongoing XMLC research.

Bibliography

L. A. Adamic and B. A. Huberman. Zipf's law and the Internet. Glottometrics, 3 (1):143–150, 2002.

A. Agarwal, A. Beygelzimer, M. Dudik, J. Langford, and H. Wallach. A Reductions Approach to Fair Classification. In J. Dy and A. Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 60–69. PMLR, 10–15 Jul 2018.

S. Agarwal. Surrogate Regret Bounds for Bipartite Ranking via Strongly Proper Losses. Journal of Machine Learning Research, 15(1):1653–1674, 2014.

R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. Multi-Label Learning with Millions of Labels: Recommending Advertiser Bid Phrases for Web pages. In 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, pages 13–24. International World Wide Web Conferences Steering Committee / ACM, 2013.

R. Babbar and B. Schölkopf. DiSMEC: Distributed sparse machines for extreme multi-label classification. In Proceedings of the tenth ACM international conference on web search and data mining, pages 721–729, New York, NY, USA, 2017. Association for Computing Machinery.

R. Babbar and B. Schölkopf. Data scarcity, robustness and extreme multi-label classification. Machine Learning, 108(8), 09 2019. doi: 10.1007/s10994-019-05791-5.

P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. Journal of the American Statistical Association, 101(473):138–156, 2006.

E. Baum and D. Haussler. What size net gives valid generalization? Advances in neural information processing systems, 1, 1988.

J. Bekker and J. Davis. Learning from positive and unlabeled data: a survey. Machine Learning, 109(4):719–760, 2020. doi: 10.1007/s10994-020-05877-5.

A. Beygelzimer, J. Langford, Y. Lifshits, G. Sorkin, and A. Strehl. Conditional Probability Tree Estimation Analysis and Algorithms. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, page 51–58, Arlington, Virginia, USA, 2009a. AUAI Press.

A. Beygelzimer, J. Langford, and P. Ravikumar. Error-Correcting Tournaments. In Algorithmic Learning Theory, 20th International Conference, ALT 2009, Porto, Portugal, October 3-5, 2009. Proceedings, volume 5809 of Lecture Notes in Computer Science, pages 247–262. Springer, 2009b.

K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. Sparse Local Embeddings for Extreme Multi-label Classification. In Advances in Neural Information Processing Systems 28, pages 730–738. Curran Associates, Inc., 2015.

K. Bhatia, K. Dahiya, H. Jain, A. Mittal, Y. Prabhu, and M. Varma. The extreme classification repository: Multi-label datasets and code, 2016. URL `http://manikvarma.org/downloads/XC/XMLRepository.html`.

G. Blanchard, G. Lee, and C. Scott. Semi-Supervised Novelty Detection. Journal of Machine Learning Research, 11(99):2973–3009, 2010.

R. Busa-Fekete, B. Szörényi, K. Dembczyński, and E. Hüllermeier. Online F-Measure Optimization. In Advances in Neural Information Processing Systems 28, pages 595–603. Curran Associates, Inc., 2015.

R. Busa-Fekete, K. Dembczyński, A. Golovnev, K. Jasinska, M. Kuznetsov, M. Sviridenko, and C. Xu. On the computational complexity of the probabilistic label tree algorithms. CoRR, abs/1906.00294, 2019.

C. Calauzenes, N. Usunier, and P. Gallinari. On the (non-) existence of convex, calibrated surrogate losses for ranking. Advances in Neural Information Processing Systems, 25, 2012.

W. Chang, H. Yu, K. Zhong, Y. Yang, and I. S. Dhillon. Taming Pretrained Transformers for Extreme Multi-label Text Classification. In R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, editors, KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020, pages 3163–3171. ACM, 2020.

W.-C. Chang, D. Jiang, H.-F. Yu, C.-H. Teo, J. Zhang, K. Zhong, K. Kolluri, Q. Hu, N. Shandilya, V. Ievgrafov, J. Singh, and I. S. Dhillon. Extreme Multi-label Learning for Semantic Matching in Product Search. In Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2021.

E. Chien, J. Zhang, C.-J. Hsieh, J.-Y. Jiang, W.-C. Chang, O. Milenkovic, and H.-F. Yu. PINA: Leveraging Side Information in eXtreme Multi-label Classification via

Predicted Instance Neighborhood Aggregation. arXiv preprint arXiv:2305.12349, 2023.

A. E. Choromanska and J. Langford. Logarithmic Time Online Multiclass prediction. In Advances in Neural Information Processing Systems 28, pages 55–63. Curran Associates, Inc., 2015.

P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In Proceedings of the fourth ACM conference on Recommender systems, pages 39–46, 2010.

K. Dahiya, A. Agarwal, D. Saini, K. Gururaj, J. Jiao, A. Singh, S. Agarwal, P. Kar, and M. Varma. SiameseXML: Siamese Networks meet Extreme Classifiers with 100M Labels. In International Conference on Machine Learning, pages 2330–2340. PMLR, 2021.

K. Dahiya, N. Gupta, D. Saini, A. Soni, Y. Wang, K. Dave, J. Jiao, G. K, P. Dey, A. Singh, et al. NGAME: Negative mining-aware mini-batching for extreme classification. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pages 258–266, 2023a.

K. Dahiya, S. Yadav, S. Sondhi, D. Saini, S. Mehta, J. Jiao, S. Agarwal, P. Kar, and M. Varma. Deep encoders with auxiliary parameters for extreme classification. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 358–367, 2023b.

O. Dekel and O. Shamir. Multiclass-Multilabel Classification with More Classes than Examples. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, volume 9 of JMLR Proceedings, pages 137–144, Chia Laguna Resort, Sardinia, Italy, 2010. PMLR.

K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier. On Label Dependence and Loss Minimization in Multi-Label Classification. Machine Learning, 88(1-2):5–45, 2012.

K. Dembczyński, W. Kotłowski, O. Koyejo, and N. Natarajan. Consistency Analysis for Binary Classification Revisited. In Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 961–969, International Convention Centre, Sydney, Australia, 2017. PMLR.

K. Dembczyński, W. Kotłowski, W. Waegeman, R. Busa-Fekete, and E. Hüllermeier. Consistency of probabilistic classifier trees. In ECML PKDD 2016 : machine learning and knowledge discovery in databases, volume 9852 of Lecture Notes in Computer Science, pages 511–526. Springer, 2016.

K. J. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier. An Exact Algorithm for F-Measure Maximization. In Advances in Neural Information Processing Systems 24, pages 1404–1412. Curran Associates, Inc., 2011.

J. Deng, S. Satheesh, A. C. Berg, and F. Li. Fast and Balanced: Efficient Label Tree Learning for Large Scale Object Recognition. In Advances in Neural Information Processing Systems 24, pages 567–575. Curran Associates, Inc., 2011.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), pages 4171–4186, 2019.

J. C. Duchi, L. W. Mackey, and M. I. Jordan. On the Consistency of Ranking Algorithms. In Proceedings of the 27th International Conference on International Conference on Machine Learning, page 327–334, 2010.

C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '08, pages 213–220, New York, NY, USA, aug 2008. Association for Computing Machinery. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401920.

I. Evron, E. Moroshko, and K. Crammer. Efficient loss-based decoding on graphs for extreme classification. Advances in Neural Information Processing Systems, 31, 2018.

S. Exchange. Continuous linear image of closed, bounded, and convex set of a Hilbert Space is compact. URL `https://math.stackexchange.com/q/908121`. Accessed: 2023-09-27.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research, 9:1871–1874, 2008.

J. Fox. Applied regression analysis, linear models, and related methods. Sage, 1997.

M. Frank and P. Wolfe. An algorithm for quadratic programming. Naval Research Logistics Quarterly, 3(1-2):95–110, 1956. doi: 10.1002/nav.3800030109.

D. Gillick, A. Presta, and G. S. Tomar. End-to-end retrieval in continuous space. arXiv preprint arXiv:1811.08008, 2018.

J. Goodman. Classes for fast maximum entropy training. In 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221), volume 1, pages 561–564 vol.1, 2001. doi: 10.1109/ICASSP.2001.940893.

C. Guo, A. Mousavi, X. Wu, D. N. Holtmann-Rice, S. Kale, S. Reddi, and S. Kumar. Breaking the Glass Ceiling for Embedding-Based Classifiers for Large Output Spaces. In Advances in Neural Information Processing Systems, volume 32, pages 4943–4953. Curran Associates, Inc., 2019.

N. Gupta, D. Khatri, A. S. Rawat, S. Bhojanapalli, P. Jain, and I. Dhillon. Dual-encoders for extreme multi-label classification. arXiv preprint arXiv:2310.10636, 2023.

M. Hanif and K. R. W. Brewer. Sampling with unequal probabilities without replacement: a review. International Statistical Review, 48:317–335, 1980.

C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. Nature, 585 (7825):357–362, Sept. 2020. doi: 10.1038/s41586-020-2649-2.

S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In International conference on machine learning, pages 427–435. PMLR, 2013.

H. Jain, Y. Prabhu, and M. Varma. Extreme Multi-Label Loss Functions for Recommendation, Tagging, Ranking & Other Missing Label Applications. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, page 935–944, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939756.

H. Jain, V. Balasubramanian, B. Chunduri, and M. Varma. Slice: Scalable Linear Extreme Classifiers Trained on 100 Million Labels for Related Searches. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pages 528–536, Melbourne VIC Australia, Jan. 2019. ACM. ISBN 978-1-4503-5940-5. doi: 10.1145/3289600.3290979.

K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems (TOIS), 20(4):422–446, 2002.

K. Jasinska and K. Dembczyński. Bayes optimal prediction for NDCG@k in extreme amulti-label classification. In From Multiple Criteria Decision Aid to Preference Learning Workshop, 2018.

K. Jasinska and N. Karampatziakis. Log-time and Log-space Extreme Classification. CoRR, abs/1611.01964, 2016.

K. Jasinska, K. Dembczyński, R. Busa-Fekete, K. Pfannschmidt, T. Klerx, and E. Hullermeier. Extreme F-measure Maximization using Sparse Probability Estimates. In Proceedings of The 33rd International Conference on Machine Learning, volume 48 of JMLR Workshop and Conference Proceedings, pages 1435–1444, New York, USA, 2016. PMLR.

K. Jasinska-Kobus, M. Wydmuch, K. Dembczynski, M. Kuznetsov, and R. Busa-Fekete. Probabilistic Label Trees for Extreme Multi-label Classification. 2020.

K. Jasinska-Kobus, M. Wydmuch, D. Thiruvenkatachari, and K. Dembczyński. Online probabilistic label trees. In Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, volume 130 of Proceedings of Machine Learning Research, pages 1801–1809. PMLR, 13–15 Apr. 2021.

S. Jayaram Subramanya, F. Devvrit, H. V. Simhadri, R. Krishnawamy, and R. Kadekodi. Diskann: Fast accurate billion-point nearest neighbor search on a single node. Advances in neural information processing Systems, 32, 2019.

Y. Jernite, A. Choromanska, and D. Sontag. Simultaneous Learning of Trees and Representations for Extreme Classification and Density Estimation. In Proceedings of the 34th International Conference on Machine Learning - volume 70, volume 70 of Proceedings of Machine Learning Research, page 1665–1674. JMLR.org, 2017.

T. Jiang, D. Wang, L. Sun, H. Yang, Z. Zhao, and F. Zhuang. LightXML: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 7987–7994, 2021.

T. Joachims. A Support Vector Method for Multivariate Performance Measures. In Proceedings of the 22nd International Conference on Machine Learning (ICML 2005), August 7-11, 2005, Bonn, Germany, 2005.

T. Joachims, A. Swaminathan, and T. Schnabel. Unbiased Learning-to-Rank with Biased Feedback. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, pages 5284–5288. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/738.

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of Tricks for Efficient Text Classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: volume 2, Short Papers, volume abs/1607.01759, pages 427–431, Valencia, Spain, 2017. Association for Computational Linguistics.

P. Kar, H. Narasimhan, and P. Jain. Surrogate Functions for Maximizing Precision at the Top. In Proceedings of the 32nd International Conference on Machine Learning (ICML-15), pages 189–198, 2015.

M. Kearns, S. Neel, A. Roth, and Z. S. Wu. Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness. In J. Dy and A. Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 2564–2572. PMLR, 10–15 Jul 2018.

H. J. Kelley. Gradient theory of optimal flight paths. Ars Journal, 30(10):947–954, 1960.

S. Khandagale, H. Xiao, and R. Babbar. Bonsai: diverse and shallow trees for extreme multi-label classification. Machine Learning, 109(11):2099–2119, 2020.

S. Kharbanda, A. Banerjee, E. Schultheis, and R. Babbar. CascadeXML: Rethinking Transformers for End-to-end Multi-resolution Training in Extreme Multi-label Classification. Advances in Neural Information Processing Systems, 35:2074–2087, 2022a.

S. Kharbanda, D. Gupta, E. Schultheis, A. Banerjee, V. Verma, and R. Babbar. Gandalf: Data Augmentation is all you need for Extreme Classification. 2022b.

T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

W. Kotłowski and K. Dembczyński. Surrogate regret bounds for generalized classification performance metrics. Machine Learning, 10(4):549–572, 2017.

W. Kotłowski, M. Wydmuch, E. Schultheis, R. Babbar, and K. Dembczyński. A General Online Algorithm for Optimizing Complex Performance Metrics. In Forty-first International Conference on Machine Learning, 2024.

O. O. Koyejo, N. Natarajan, P. K. Ravikumar, and I. S. Dhillon. Consistent Binary Classification with Generalized Performance Metrics. In Advances in Neural Information Processing Systems 27, pages 2744–2752. Curran Associates, Inc., 2014.

O. O. Koyejo, N. Natarajan, P. K. Ravikumar, and I. S. Dhillon. Consistent Multilabel Classification. In Advances in Neural Information Processing Systems 28, pages 3321–3329. Curran Associates, Inc., 2015.

S. I. Ktena, A. Tejani, L. Theis, P. K. Myana, D. Dilipkumar, F. Huszár, S. Yoo, and W. Shi. Addressing delayed feedback for continuous training with neural networks in CTR prediction. In Proceedings of the 13th ACM conference on recommender systems, pages 187–195, 2019.

S. K. Lam, A. Pitrou, and S. Seibert. Numba: A llvm-based python jit compiler. In Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, pages 1–6, 2015.

J. Leskovec, A. Rajaraman, and J. D. Ullman. Mining of Massive Datasets. Cambridge University Press, USA, 2nd edition, 2014. ISBN 1107077230.

D. D. Lewis. Evaluating and optimizing autonomous text classification systems. In Proceedings of the 18th Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, pages 246–254. ACM, 1995.

Y.-J. Lin and C.-J. Lin. On the Thresholding Strategy for Infrequent Labels in Multi-Label Classification. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23, page 1441–1450, New York, NY, USA, 2023. Association for Computing Machinery.

Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130, 2017.

D. Maclaurin, D. Duvenaud, and R. P. Adams. Autograd: Effortless gradients in numpy. In ICML 2015 AutoML Workshop, volume 238, page 5, 2015.

W. G. Madow. On the theory of systematic sampling, II. The Annals of Mathematical Statistics, 20(3):333–354, 1949.

Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE transactions on pattern analysis and machine intelligence, 42(4):824–836, 2018.

J. McAuley, R. Pandey, and J. Leskovec. Inferring Networks of Substitutable and Complementary Products. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, page 785–794, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336642. doi: 10.1145/2783258.2783381.

T. K. R. Medini, Q. Huang, Y. Wang, V. Mohan, and A. Shrivastava. Extreme Classification in Log Memory using Count-Min Sketch: A Case Study of Amazon Search with 50M Products. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 13265–13275. Curran Associates, Inc., 2019.

A. K. Menon, A. S. Rawat, S. Reddi, and S. Kumar. Multilabel reductions: what is my loss optimising? In Advances in Neural Information Processing Systems 32, pages 10600–10611. Curran Associates, Inc., 2019.

P. Mineiro and N. Karampatziakis. Fast Label Embeddings via Randomized Linear Algebra. In Proceedings of the 2015th European Conference on Machine Learning and Knowledge Discovery in Databases - volume Part I, volume 9284 of Lecture Notes in Computer Science, page 37–51, Gewerbestrasse 11 CH-6330, Cham (ZG), CHE, 2015. Springer.

A. Mittal, N. Sachdeva, S. Agrawal, S. Agarwal, P. Kar, and M. Varma. ECLARE: Extreme classification with label graph correlations. In Proceedings of the Web Conference 2021, pages 3721–3732, 2021.

F. Morin and Y. Bengio. Hierarchical Probabilistic Neural Network Language Model. In Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, pages 246–252. Society for Artificial Intelligence and Statistics, 2005.

S. Mukhopadhyay, S. Sahoo, and A. Sinha. k-experts - Online Policies and Fundamental Limits. In Proceedings of The 25th International Conference on Artificial Intelligence and Statistics, volume 151 of Proceedings of Machine Learning Research, pages 342–365. PMLR, 2022.

D. R. Musser. Introspective sorting and selection algorithms. Software: Practice and Experience, 27(8):983–993, 1997.

H. Narasimhan, R. Vaish, and S. Agarwal. On the Statistical Consistency of Plug-in Classifiers for Non-decomposable Performance Measures. In Advances in Neural Information Processing Systems 27, pages 1493–1501. Curran Associates, Inc., 2014.

H. Narasimhan, H. Ramaswamy, A. Saha, and S. Agarwal. Consistent Multiclass Algorithms for Complex Performance Measures. In F. Bach and D. Blei, editors, Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pages 2398–2407, Lille, France, 07–09 Jul 2015. PMLR.

H. Narasimhan, H. G. Ramaswamy, S. K. Tavker, D. Khurana, P. Netrapalli, and S. Agarwal. Consistent Multiclass Algorithms for Complex Metrics and Constraints. 2022. doi: 10.48550/ARXIV.2210.09695.

N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. Advances in Neural Information Processing Systems, 26, 2013.

N. Natarajan, O. Koyejo, P. Ravikumar, and I. Dhillon. Optimal Classification with Multivariate Losses. In M. F. Balcan and K. Q. Weinberger, editors, Proceedings of The 33rd International Conference on Machine Learning, volume 48 of Proceedings of Machine Learning Research, pages 1530–1538, New York, New York, USA, 20–22 Jun 2016a. PMLR.

N. Natarajan, O. Koyejo, P. Ravikumar, and I. Dhillon. Optimal Classification with Multivariate Losses. In Proceedings of The 33rd International Conference on Machine Learning (ICML), pages 1530–1538, 2016b.

N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari. Cost-sensitive learning with noisy labels. The Journal of Machine Learning Research, 18(1):5666–5698, 2017.

A. Niculescu-Mizil and E. Abbasnejad. Label Filters for Large Scale Multilabel Classification. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, volume 54 of Proceedings of Machine Learning Research, pages 1448–1457, Fort Lauderdale, FL, USA, 2017. PMLR.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc., 2019.

J. Pearl. Heuristics: intelligent search strategies for computer problem solving. Addison-Wesley Longman Publishing Co., Inc., USA, 1984. ISBN 0201055945.

Y. Prabhu and M. Varma. FastXML: A Fast, Accurate and Stable Tree-Classifier for Extreme Multi-Label Learning. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, page 263–272, New York, NY, USA, 2014. Association for Computing Machinery.

Y. Prabhu, A. Kag, S. Gopinath, K. Dahiya, S. Harsola, R. Agrawal, and M. Varma. Extreme multi-label learning with label features for warm-start tagging, ranking & recommendation. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pages 441–449, 2018a.

Y. Prabhu, A. Kag, S. Harsola, R. Agrawal, and M. Varma. Parabel: Partitioned Label Trees for Extreme Classification with Application to Dynamic Search Advertising. In Proceedings of the 2018 World Wide Web Conference, page 993–1002, Republic and Canton of Geneva, CHE, 2018b. International World Wide Web Conferences Steering Committee.

M. Qaraei, E. Schultheis, P. Gupta, and R. Babbar. Convex Surrogates for Unbiased Loss Functions in Extreme Classification With Missing Labels. In Proceedings of The Web Conference 2021, WWW '21, New York, NY, USA, 2021. Association for Computing Machinery. doi: 10.1145/3442381.3450139.

H. G. Ramaswamy, A. Tewari, and S. Agarwal. Consistent algorithms for multiclass classification with an abstain option. Electronic Journal of Statistics, 12(1):530 – 554, 2018. doi: 10.1214/17-EJS1388.

P. Ravikumar, A. Tewari, and E. Yang. On NDCG consistency of listwise ranking methods. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pages 618–626. JMLR Workshop and Conference Proceedings, 2011.

M. Reid and R. Williamson. Convexity of Proper Composite Binary Losses. In Y. W. Teh and M. Titterington, editors, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, volume 9 of Proceedings of Machine Learning Research, pages 637–644, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

S. J. Russell and P. Norvig. Artificial Intelligence: a modern approach. Pearson, 3 edition, 2009.

D. Saini, A. K. Jain, K. Dave, J. Jiao, A. Singh, R. Zhang, and M. Varma. GalaXC: Graph neural networks with labelwise attention for extreme classification. In Proceedings of the Web Conference 2021, pages 3733–3744, 2021.

Y. Saito, S. Yaginuma, Y. Nishino, H. Sakata, and K. Nakata. Unbiased Recommender Learning from Missing-Not-At-Random Implicit Feedback. In Proceedings of the 13th International Conference on Web Search and Data Mining,

WSDM '20, page 501–509, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368223. doi: 10.1145/3336191.3371783.

E. Schultheis and R. Babbar. Unbiased Loss Functions for Multilabel Classification with Missing Labels. CoRR, abs/2109.11282, 2021.

E. Schultheis, M. Wydmuch, R. Babbar, and K. Dembczyński. On Missing Labels, Long-tails and Propensities in Extreme Multi-label Classification. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22, page 1547–1557, New York, NY, USA, 2022. Association for Computing Machinery.

E. Schultheis, M. Wydmuch, W. Kotłowski, R. Babbar, and K. Dembczyński. Generalized test utilities for long-tail performance in extreme multi-label classification. In Advances in Neural Information Processing Systems, volume 36, pages 22269–22303. Curran Associates, Inc., 2023.

E. Schultheis, W. Kotłowski, M. Wydmuch, R. Babbar, S. Borman, and K. Dembczyński. Consistent algorithms for multi-label classification with macro-at-k metrics. In The Twelfth International Conference on Learning Representations, 2024.

A. Shrivastava and P. Li. Improved Asymmetric Locality Sensitive Hashing (ALSH) for Maximum Inner Product Search (MIPS). In Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, page 812–821, Arlington, Virginia, USA, 2015. AUAI Press.

S. Singh and J. T. Khim. Optimal Binary Classification Beyond Accuracy. In Advances in Neural Information Processing Systems, volume 35, pages 18226–18240. Curran Associates, Inc., 2022.

L. Song, P. Pan, K. Zhao, H. Yang, Y. Chen, Y. Zhang, Y. Xu, and R. Jin. Large-Scale Training System for 100-Million Classification at Alibaba. KDD '20, page 2909–2930, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403342.

Y. Tagami. AnnexML: Approximate Nearest Neighbor Search for Extreme Multi-label Classification. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017, pages 455–464. ACM, 2017a.

Y. Tagami. AnnexML: Approximate nearest neighbor search for extreme multi-label classification. In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pages 455–464, 2017b.

P. Teisseyre, J. Mielniczuk, and M. Łazęcka. Different Strategies of Fitting Logistic Regression for Positive and Unlabelled Data. In V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, and J. Teixeira, editors, Computational Science – ICCS 2020, pages 3–17, Cham, 2020. Springer International Publishing. ISBN 978-3-030-50423-6.

A. Tewari and P. L. Bartlett. On the Consistency of Multiclass Classification Methods. Journal of Machine Learning Research, 8(5), 2007.

B. Van Rooyen and R. C. Williamson. A theory of learning with corrupted labels. The Journal of Machine Learning Research, 18(1):8501–8550, Jan. 2017. ISSN 1532-4435.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is All you Need. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

S. Vijayanarasimhan, J. Shlens, R. Monga, and J. Yagnik. Deep Networks With Large Output Spaces. CoRR, abs/1412.7479, 2014.

P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research, 11(12), 2010.

W. Waegeman, K. Dembczyński, A. Jachnik, W. Cheng, and E. Hüllermeier. On the Bayes-Optimality of F-Measure Maximizers. Journal of Machine Learning Research, 15(1):3513–3568, 2014.

B. Wang, L. Chen, W. Sun, K. Qin, K. Li, and H. Zhou. Ranking-based autoencoder for extreme multi-label classification. arXiv preprint arXiv:1904.05937, 2019a.

X. Wang, R. Li, B. Yan, and O. Koyejo. Consistent classification with generalized metrics. arXiv preprint arXiv:1908.09057, 2019b.

T. Wei and Y.-F. Li. Does Tail Label Help for Large-Scale Multi-Label Learning? IEEE Transactions on Neural Networks and Learning Systems, 31(7):2315–2324, 2020. doi: 10.1109/TNNLS.2019.2935143.

T. Wei, W.-W. Tu, Y.-F. Li, and G.-P. Yang. Towards robust prediction on tail labels. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pages 1812–1820, 2021.

J. Weston, A. Makadia, and H. Yee. Label Partitioning For Sublinear Ranking. In Proceedings of the 30th International Conference on Machine Learning, volume 28 of JMLR Workshop and Conference Proceedings, pages 181–189, Atlanta, Georgia, USA, 2013. PMLR.

M. Wydmuch, K. Jasinska, M. Kuznetsov, R. Busa-Fekete, and K. Dembczyński. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In Advances in Neural Information Processing Systems, volume 31, pages 6358–6368. Curran Associates, Inc., 2018.

M. Wydmuch, K. Jasinska-Kobus, R. Babbar, and K. Dembczyński. Propensity-Scored Probabilistic Label Trees. In Proceedings of the 44th International

ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21, page 2252–2256, New York, NY, USA, 2021. Association for Computing Machinery.

F. Yang and S. Koyejo. On the consistency of top-k surrogate losses. In International Conference on Machine Learning, pages 10727–10735. PMLR, 2020.

L. Yang, Y. Cui, Y. Xuan, C. Wang, S. Belongie, and D. Estrin. Unbiased Offline Recommender Evaluation for Missing-Not-at-Random Implicit Feedback. In Proceedings of the 12th ACM Conference on Recommender Systems, RecSys '18, page 279–287, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450359016. doi: 10.1145/3240323.3240355.

Y. Yang. A Study of Thresholding Strategies for Text Categorization. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '01, page 137–145, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581133316.

S. Yasui, G. Morishita, F. Komei, and M. Shibata. A feedback shift correction in predicting conversion rates under delayed feedback. In Proceedings of The Web Conference 2020, pages 2740–2746, 2020.

H. Ye, Z. Chen, D.-H. Wang, and B. Davison. Pretrained generalized autoregressive model with adaptive probabilistic label clusters for extreme multi-label text classification. In International Conference on Machine Learning, pages 10809–10819. PMLR, 2020.

N. Ye, K. M. A. Chai, W. S. Lee, and H. L. Chieu. Optimizing F-Measures: A Tale of Two Approaches. In Proceedings of the 29th International Coference on International Conference on Machine Learning, page 1555–1562, Madison, WI, USA, 2012. Omnipress.

C.-K. Yeh, W.-C. Wu, W.-J. Ko, and Y.-C. F. Wang. Learning deep latent space for multi-label classification. In Proceedings of the AAAI conference on artificial intelligence, volume 31, 2017.

I. E. Yen, X. Huang, W. Dai, P. Ravikumar, I. Dhillon, and E. Xing. PPDsparse: A Parallel Primal-Dual Sparse Method for Extreme Classification. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, page 545–553. Association for Computing Machinery, 2017.

R. You, Z. Zhang, Z. Wang, S. Dai, H. Mamitsuka, and S. Zhu. AttentionXML: Label Tree-based Attention-Aware Deep Model for High-Performance Extreme Multi-Label Text Classification. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 5820–5830. Curran Associates, Inc., 2019a.

R. You, Z. Zhang, Z. Wang, S. Dai, H. Mamitsuka, and S. Zhu. AttentionXML: Label tree-based attention-aware deep model for high-performance extreme

multi-label text classification. Advances in neural information processing systems, 32, 2019b.

H.-F. Yu, P. Jain, P. Kar, and I. Dhillon. Large-scale Multi-label Learning with Missing Labels. In Proceedings of the 31st International Conference on Machine Learning, volume 32 of JMLR Workshop and Conference Proceedings, pages 593–601, Bejing, China, 2014. PMLR.

H.-F. Yu, K. Zhong, J. Zhang, W.-C. Chang, and I. S. Dhillon. Pecos: Prediction for enormous and correlated output spaces. Journal of Machine Learning Research, 23(98):1–32, 2022.

M. Yuan and M. Wegkamp. Classification Methods with Reject Option Based on Convex Risk Minimization. J. Mach. Learn. Res., 11:111–130, mar 2010. ISSN 1532-4435.

J. Zhang, W.-c. Chang, H.-f. Yu, and I. Dhillon. Fast multi-resolution transformer fine-tuning for extreme multi-label text classification. Advances in Neural Information Processing Systems, 34, 2021.

R. Zhang, Y.-S. Wang, Y. Yang, D. Yu, T. Vu, and L. Lei. Long-tailed extreme multi-label text classification with generated pseudo label descriptions. arXiv preprint arXiv:2204.00958, 2022.

T. Zhang. Statistical analysis of some multi-category large margin classification methods. Journal of Machine Learning Research, 5(Oct):1225–1251, 2004.

W. Zhang, J. Yan, X. Wang, and H. Zha. Deep extreme multi-label learning. In Proceedings of the 2018 ACM on international conference on multimedia retrieval, pages 100–107, 2018.

Z. Zhu, Y. He, Y. Zhang, and J. Caverlee. Unbiased Implicit Recommendation and Propensity Estimation via Combinational Joint Learning. In Fourteenth ACM Conference on Recommender Systems, RecSys '20, page 551–556, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375832. doi: 10.1145/3383313.3412210.

J. Zhuo, Z. Xu, W. Dai, H. Zhu, H. Li, J. Xu, and K. Gai. Learning Optimal Tree Models under Beam Search. In Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, 2020. PMLR.

D. Zong and S. Sun. GNN-XML: graph neural networks for extreme multi-label text classification. arXiv preprint arXiv:2012.05860, 2020.

A
Full proofs

In this appendix, we provide full proofs of theorems and lemmas from the main text that were presented as sketches in the main text or completely omitted. The proofs are divided into sections, each section corresponds to a chapter from the main text, and presented in the same order as they appear in the main text.

A.1 Chapter 3

A.1.1 Equivalence of optimal classifiers for precision@k and recall@k under labels independence

Theorem 3.3.1. *Given conditionally independent labels, $\eta_j(\boldsymbol{x})$ and $\eta'_j(\boldsymbol{x})$, $j \in [m]$ induce the same order of labels.*

Proof. To prove the theorem, it suffices to show that for conditionally independent labels, the order of labels induced by the marginal probabilities $\eta_j(\boldsymbol{x})$ is the same as the order induced by the values of $\eta'_j(\boldsymbol{x})$:

$$\eta'_j(\boldsymbol{x}) = \mathbb{P}'[y_j = 1 \,|\, \boldsymbol{x}] = \sum_{\boldsymbol{y} \in \mathcal{Y}} \frac{y_j}{\|\boldsymbol{y}\|_1} \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}]. \tag{A.1}$$

In other words, for any two labels $i, j \in [m]$, $i \neq j$, $\eta_i(\boldsymbol{x}) \geqslant \eta_j(\boldsymbol{x}) \Leftrightarrow \eta'_i(\boldsymbol{x}) \geqslant \eta'_j(\boldsymbol{x})$.

Let assume that $\eta_i(\boldsymbol{x}) \geqslant \eta_j(\boldsymbol{x})$. The summation over all $\boldsymbol{y}$ in (A.1) can be written in the following way:

$$\eta'_j(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \mathcal{Y}} y_j N(\boldsymbol{y}) \mathbb{P}[\boldsymbol{y} | \boldsymbol{x}]\,, \tag{A.2}$$

where $N(\boldsymbol{y}) = (\|\boldsymbol{y}\|_1)^{-1}$ is a value that depends only on the number of positive labels in $\boldsymbol{y}$. In this summation, we consider four subsets of $\mathcal{Y}$, creating a partition

of this set:

$$\mathcal{S}_{i,j}^{u,w} = \{\boldsymbol{y} \in \mathcal{Y} : y_i = u \wedge y_j = w\}, \quad u, w \in \{0, 1\}. \tag{A.3}$$

The subset $\mathcal{S}_{i,j}^{0,0}$ does not play any role because $y_i = y_j = 0$ and therefore do not contribute to the final sum. Then (A.1) can be written in the following way for the i-th and j-th label:

$$\eta_i'(\boldsymbol{x}) = \sum_{\boldsymbol{y}:\mathcal{S}_{i,j}^{1,0}} N(\boldsymbol{y})\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}] + \sum_{\boldsymbol{y}\in\mathcal{S}_{i,j}^{1,1}} N(\boldsymbol{y})\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}] \tag{A.4}$$

$$\eta_j'(\boldsymbol{x}) = \sum_{\boldsymbol{y}:\mathcal{S}_{i,j}^{0,1}} N(\boldsymbol{y})\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}] + \sum_{\boldsymbol{y}\in\mathcal{S}_{i,j}^{1,1}} N(\boldsymbol{y})\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}] \tag{A.5}$$

The contribution of elements from $\mathcal{S}_{i,j}^{1,1}$ is equal for both $\eta_i'(\boldsymbol{x})$ and $\eta_j'(\boldsymbol{x})$. It is so because the value of $N(\boldsymbol{y})\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}]$ is the same for all $\boldsymbol{y} \in \mathcal{S}_{i,j}^{1,1}$: the conditional joint probabilities $\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}]$ are fixed and they are multiplied by the same factors $N(\boldsymbol{y})$.

Consider now the contributions of $\mathcal{S}_{i,j}^{1,0}$ and $\mathcal{S}_{i,j}^{0,1}$ to the relevant sums. By the definition of $\mathcal{Y}$, $\mathcal{S}_{i,j}^{1,0}$, and $\mathcal{S}_{i,j}^{0,1}$, there exists bijection $b_{i,j} : \mathcal{S}_{i,j}^{1,0} \to \mathcal{S}_{i,j}^{0,1}$, such that for each $\boldsymbol{y}' \in \mathcal{S}_{i,j}^{1,0}$ there exists $\boldsymbol{y}'' \in \mathcal{S}_{i,j}^{0,1}$ equal to $\boldsymbol{y}'$ except on the i-th and the j-th position.

Notice that because of the conditional independence assumption the joint probabilities of elements in $\mathcal{S}_{i,j}^{1,0}$ and $\mathcal{S}_{i,j}^{0,1}$ are related to each other. Let $\boldsymbol{y}'' = b_{i,j}(\boldsymbol{y}')$, where $\boldsymbol{y}' \in \mathcal{S}_{i,j}^{1,0}$ and $\boldsymbol{y}'' \in \mathcal{S}_{i,j}^{0,1}$. The joint probabilities are:

$$\mathbb{P}[\boldsymbol{y}'|\boldsymbol{x}] = \eta_i(\boldsymbol{x})(1 - \eta_j(\boldsymbol{x})) \prod_{l\in\mathcal{L}\setminus\{i,j\}} \eta_l(\boldsymbol{x})^{y_l}(1 - \eta_l(\boldsymbol{x}))^{1-y_l} \tag{A.6}$$

and

$$\mathbb{P}[\boldsymbol{y}''|\boldsymbol{x}] = (1 - \eta_i(\boldsymbol{x}))\eta_j(\boldsymbol{x}) \prod_{l\in\mathcal{L}\setminus\{i,j\}} \eta_l(\boldsymbol{x})^{y_l}(1 - \eta_l(\boldsymbol{x}))^{1-y_l}. \tag{A.7}$$

One can easily notice the relation between these probabilities:

$$\mathbb{P}[\boldsymbol{y}'|\boldsymbol{x}] = \eta_i(\boldsymbol{x})(1 - \eta_j(\boldsymbol{x}))q_{i,j} \quad \text{and} \quad \mathbb{P}[\boldsymbol{y}''|\boldsymbol{x}] = (1 - \eta_i(\boldsymbol{x}))\eta_j(\boldsymbol{x})q_{i,j}, \tag{A.8}$$

where $q_{i,j} = \prod_{l\in\mathcal{L}\setminus\{i,j\}} \eta_l(\boldsymbol{x})^{y_l}(1 - \eta_l(\boldsymbol{x}))^{1-y_l} \geqslant 0$. Consider now the difference of these two probabilities:

$$\mathbb{P}[\boldsymbol{y}'|\boldsymbol{x}] - \mathbb{P}[\boldsymbol{y}''|\boldsymbol{x}] = \eta_i(\boldsymbol{x})(1 - \eta_j(\boldsymbol{x}))q_{i,j} - (1 - \eta_i(\boldsymbol{x}))\eta_j(\boldsymbol{x})q_{i,j} \tag{A.9}$$

$$= q_{i,j}(\eta_i(\boldsymbol{x})(1 - \eta_j(\boldsymbol{x})) - (1 - \eta_i(\boldsymbol{x}))\eta_j(\boldsymbol{x})) \tag{A.10}$$

$$= q_{i,j}(\eta_i(\boldsymbol{x}) - \eta_j(\boldsymbol{x})). \tag{A.11}$$

From the above we see that $\eta_i(\boldsymbol{x}) \geqslant \eta_j(\boldsymbol{x}) \Rightarrow \mathbb{P}[\boldsymbol{y}'|\boldsymbol{x}] \geqslant \mathbb{P}[\boldsymbol{y}''|\boldsymbol{x}]$. Due to the properties of the bijection $b_{i,j}$, the number of positive labels in $\boldsymbol{y}'$ and $\boldsymbol{y}''$ is the same and $N(\boldsymbol{y}') = N(\boldsymbol{y}'')$, therefore we also get $\eta_i(\boldsymbol{x}) \geqslant \eta_j(\boldsymbol{x}) \Rightarrow \sum_{\boldsymbol{y}:\mathcal{S}_{i,j}^{1,0}} N(\boldsymbol{y})\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}] \geqslant \sum_{\boldsymbol{y}:\mathcal{S}_{i,j}^{0,1}} N(\boldsymbol{y})\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}]$, which by A.1.1 and A.1.1 gives us finally $\eta_i(\boldsymbol{x}) \geqslant \eta_j(\boldsymbol{x}) \Rightarrow \eta_i'(\boldsymbol{x}) \geqslant \eta_j'(\boldsymbol{x})$.

The implication in the other side, i.e., $\eta_i(\boldsymbol{x}) \geqslant \eta_j(\boldsymbol{x}) \Leftarrow \mathbb{P}[\boldsymbol{y}'|\boldsymbol{x}] \geqslant \mathbb{P}[\boldsymbol{y}''|\boldsymbol{x}]$

holds obviously for $q_{i,j} > 0$. For $q_{i,j} = 0$, we can notice, however, that $\mathbb{P}[\boldsymbol{y}'|\boldsymbol{x}]$ and $\mathbb{P}[\boldsymbol{y}''|\boldsymbol{x}]$ do not contribute to the appropriate sums as they are zero, and therefore we can follow a similar reasoning as above, concluding that $\eta_i(\boldsymbol{x}) \geqslant \eta_j(\boldsymbol{x}) \Leftarrow \eta'_i(\boldsymbol{x}) \geqslant \eta'_j(\boldsymbol{x})$.

Thus for conditionally independent labels, the order of labels induced by marginal probabilities $\eta_j(\boldsymbol{x})$ is equal to the order induced by $\eta'_j(\boldsymbol{x})$. As the precision@k is optimized by k labels with the highest marginal probabilities, we have that prediction consisted of k labels with highest $\eta'_j(\boldsymbol{x})$ has zero regret for precision@k, and vice versa for recall@k. □

A.1.2 Regret for optimal classifier for recall@k under probability estimation error

Theorem 3.3.2. *For any distribution* $\mathbb{P}(\boldsymbol{y} \,|\, \boldsymbol{x})$ *and the classifier* $\mathcal{H}^{@k} \ni \boldsymbol{h}^{@k} =$ select-top-$k(\widehat{\boldsymbol{\eta}}'(\boldsymbol{x}))$*, the following holds:*

$$\mathrm{Reg}_{\mathrm{R@}k}\left(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}\right) \leqslant 2k \max_{j\in[m]} \left|\eta'_j(\boldsymbol{x}) - \widehat{\eta}'_j(\boldsymbol{x})\right| . \tag{3.23}$$

Proof. The conditional regret for recall@k is:

$$\begin{aligned}\mathrm{Reg}_{\mathrm{R@}k}(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}) &= \Phi_{\mathrm{R@}k}(\boldsymbol{h}^{@k,\star} \,|\, \boldsymbol{x}) - \Phi_{\mathrm{R@}k}(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}) \\ &= \sum_{j=1}^{m} \eta'_j(\boldsymbol{x}) y_j^\star - \sum_{j=1}^{m} \eta'_j(\boldsymbol{x}) \widehat{y}_j \,. \end{aligned} \tag{A.12}$$

Let us add and subtract the following two terms:

$$\sum_{j=1}^{m} \widehat{\eta}'_j(\boldsymbol{x}) y_j^\star \,, \qquad \sum_{j=1}^{m} \widehat{\eta}'_j(\boldsymbol{x}) \widehat{y}_j \,, \tag{A.13}$$

from the regret and reorganize the expression in the following way:

$$\begin{aligned}\mathrm{Reg}_{\mathrm{R@}k}(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}) = &\underbrace{\sum_{j=1}^{m} \eta'_j(\boldsymbol{x}) y_j^\star - \sum_{j=1}^{m} \widehat{\eta}'_j(\boldsymbol{x}) y_j^\star}_{\leqslant \sum_{j=1}^{m} \left|\eta'_j(\boldsymbol{x}) - \widehat{\eta}'_j(\boldsymbol{x})\right| y_j^\star} + \underbrace{\sum_{j=1}^{m} \widehat{\eta}'_j(\boldsymbol{x}) \widehat{y}_j - \sum_{j=1}^{m} \eta'_j(\boldsymbol{x}) \widehat{y}_j}_{\leqslant \sum_{j=1}^{m} \left|\eta'_j(\boldsymbol{x}) - \widehat{\eta}'_j(\boldsymbol{x})\right| \widehat{y}_j} \\ &+ \underbrace{\sum_{j=1}^{m} \widehat{\eta}'_j(\boldsymbol{x}) y_j^\star - \sum_{j=1}^{m} \widehat{\eta}'_j(\boldsymbol{x}) \widehat{y}_j}_{\leqslant 0} \\ \leqslant &\sum_{j=1}^{m} \left|\eta'_j(\boldsymbol{x}) - \widehat{\eta}'_j(\boldsymbol{x})\right| y_j^\star + \sum_{j=1}^{m} \left|\eta'_j(\boldsymbol{x}) - \widehat{\eta}'_j(\boldsymbol{x})\right| \widehat{y}_j \,. \end{aligned} \tag{A.14}$$

Next we bound each L_1 error, $\eta'_j(\boldsymbol{x}) - \widehat{\eta}'_j(\boldsymbol{x})$ by $\max_{j\in[m]} \eta'_j(\boldsymbol{x}) - \widehat{\eta}'_j(\boldsymbol{x})$. There are at most $\|\boldsymbol{y}^{@k,\star} \vee \widehat{\boldsymbol{y}}^{@k}\|_1 \leqslant 2k$ such terms that stay positive. Therefore:

$$\mathrm{Reg}_{\mathrm{R@}k}(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}) \leqslant 2k \max_{j\in[m]} \left|\eta'_j(\boldsymbol{x}) - \widehat{\eta}'_j(\boldsymbol{x})\right| . \tag{A.15}$$

□

A.1.3 Regret for optimal classifier for wDCG@k and wnDCG@k under probability estimation error

Theorem 3.4.1. *For any distribution $\mathbb{P}(\boldsymbol{y} \mid \boldsymbol{x})$, vector of label gains $\boldsymbol{g}$, vector of discount factors $\boldsymbol{d}$, and the classifier $\mathcal{H}^{@k} \ni \boldsymbol{h}^{@k} = \text{select-top-}k(g \odot \widehat{\boldsymbol{\eta}}(\boldsymbol{x}))$, the following holds:*

$$\begin{aligned} \operatorname{Reg}_{\text{wDCG@}k}\left(\boldsymbol{h}^{@k} \mid \boldsymbol{x}\right) &\leqslant 2 \sum_{i}^{k} d_i \max_{j \in [m]} g_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| , \\ \operatorname{Reg}_{\text{wnDCG@}k}\left(\boldsymbol{h}^{@k} \mid \boldsymbol{x}\right) &\leqslant 2 \max_{j \in [m]} g_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| . \end{aligned} \tag{3.32}$$

Proof. The conditional regret for wDCG@k is:

$$\begin{aligned} \operatorname{Reg}_{\text{wDCG@}k}(\boldsymbol{h}^{@k} \mid \boldsymbol{x}) &= \Phi_{\text{wDCG@}k}(\boldsymbol{h}^{@k,\star} \mid \boldsymbol{x}) - \Phi_{\text{wDCG@}k}(\boldsymbol{h}^{@k} \mid \boldsymbol{x}) \\ &= \sum_{j=1}^{m} d_{\operatorname{rank}(\boldsymbol{\eta}(\boldsymbol{x}),j)} g_j \eta_j(\boldsymbol{x}) y_j^\star - \sum_{j=1}^{m} d_{\operatorname{rank}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}),j)} g_j \eta_j(\boldsymbol{x}) \widehat{y}_j . \end{aligned} \tag{A.16}$$

To prove that, let us add and subtract the following two terms:

$$\sum_{j=1}^{m} d_{\operatorname{rank}(\boldsymbol{\eta}(\boldsymbol{x}),j)} g_j \widehat{\eta}_j(\boldsymbol{x}) y_j^\star , \qquad \sum_{j=1}^{m} d_{\operatorname{rank}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}),j)} g_j \widehat{\eta}_j(\boldsymbol{x}) \widehat{y}_j , \tag{A.17}$$

from the regret and reorganize the expression in the following way:

$$\begin{aligned} \operatorname{Reg}_{\text{wDCG@}k}(\boldsymbol{h}^{@k} \mid \boldsymbol{x}) = & \underbrace{\sum_{j=1}^{m} d_{\operatorname{rank}(\boldsymbol{\eta}(\boldsymbol{x}),j)} g_j \eta_j(\boldsymbol{x}) y_j^\star - \sum_{j=1}^{m} d_{\operatorname{rank}(\boldsymbol{\eta}(\boldsymbol{x}),j)} g_j \widehat{\eta}_j(\boldsymbol{x}) y_j^\star}_{\leqslant \sum_{j=1}^{m} d_{\operatorname{rank}(\boldsymbol{\eta}(\boldsymbol{x}),j)} g_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| y_j^\star} \\ & + \underbrace{\sum_{j=1}^{m} d_{\operatorname{rank}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}),j)} g_j \widehat{\eta}_j(\boldsymbol{x}) \widehat{y}_j - \sum_{j=1}^{m} d_{\operatorname{rank}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}),j)} g_j \eta_j(\boldsymbol{x}) \widehat{y}_j}_{\leqslant \sum_{j=1}^{m} d_{\operatorname{rank}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}),j)} g_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| \widehat{y}_j} \\ & + \underbrace{\sum_{j=1}^{m} d_{\operatorname{rank}(\boldsymbol{\eta}(\boldsymbol{x}),j)} g_j \widehat{\eta}_j(\boldsymbol{x}) y_j^\star - \sum_{j=1}^{m} d_{\operatorname{rank}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}),j)} g_j \widehat{\eta}_j(\boldsymbol{x}) \widehat{y}_j}_{\leqslant 0} \\ \leqslant & \sum_{j=1}^{m} d_{\operatorname{rank}(\boldsymbol{\eta}(\boldsymbol{x}),j)} g_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| y_j^\star \\ & + \sum_{j=1}^{m} d_{\operatorname{rank}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}),j)} g_j \left|\eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x})\right| \widehat{y}_j . \end{aligned} \tag{A.18}$$

Next we bound each L_1 error, $g_j \left|\eta_j'(\boldsymbol{x}) - \widehat{\eta}_j'(\boldsymbol{x})\right|$ by $\max_{j \in [m]} g_j \left|\eta_j'(\boldsymbol{x}) - \widehat{\eta}_j'(\boldsymbol{x})\right|$.

Additionally, let us notice that both $\boldsymbol{y}^{@k,\star}$ and $\widehat{\boldsymbol{y}}^{@k}$ contains exactly k positive labels and each label appears at only one, unique rank. Therefore $\sum_{j=1}^{m} d_{\operatorname{rank}(\boldsymbol{\eta}(\boldsymbol{x}),j)} y_j^\star = \sum_{j=1}^{m} d_{\operatorname{rank}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}),j)} \widehat{y}_j = \sum_{i=1}^{k} d_i$ and we end up with the final bound for wDCG@k:

$$\operatorname{Reg}_{\mathrm{wDCG@}k}(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}) \leqslant 2 \sum_{i}^{k} d_i \max_{j \in [m]} g_j \left| \eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x}) \right| . \tag{A.19}$$

wnDCG@k is wDCG@k divided by constant iDCG@k (3.27) that cancels $\sum_i^k d_i$ term resulting with the final bound:

$$\operatorname{Reg}_{\mathrm{wnDCG@}k}(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}) \leqslant 2 \max_{j \in [m]} g_j \left| \eta_j(\boldsymbol{x}) - \widehat{\eta}_j(\boldsymbol{x}) \right| . \tag{A.20}$$

□

A.2 Chapter 4

A.2.1 Unbiased DCG at k

Theorem A.2.1 (Unbiased DCG under EIU framework)**.** *Under Definition 4.1.1 and missing labels model*

$$\widetilde{\mathrm{DCG}}@k(\boldsymbol{x}, \breve{\boldsymbol{y}}, \boldsymbol{h}^{@k}(\boldsymbol{x})) \coloneqq \sum_{j=1}^{m} \frac{\breve{y}_j \widehat{y}_j d_{\operatorname{rank}(j)}}{p_j(\boldsymbol{x})} , \tag{A.21}$$

is an unbiased estimate of DCG@k (3.25)*:*

$$\mathrm{DCG}@k(\boldsymbol{y}, \boldsymbol{h}^{@k}(\boldsymbol{x})) \coloneqq \sum_{j=1}^{m} y_j \widehat{y}_j d_{\operatorname{rank}(j)} ,$$

Proof. Let $\mathcal{S}$ be $\{0,1\}^{m-1}$ and $\boldsymbol{s}^j = [y_1, \ldots, y_{j-1}, y_{j+1}, \ldots, y_m] \in \mathcal{S}$ be a label vector without label j. Then, we have:

$$\begin{aligned}
\Phi_{\mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}]}(\boldsymbol{h}^{@k} \,|\, \boldsymbol{x}) &= \mathbb{E}_{\boldsymbol{y} \sim \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}]} \Big[\mathrm{DCG}@k(\boldsymbol{y}, \boldsymbol{h}^{@k}(\boldsymbol{x})) \Big] = \sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{P}[\boldsymbol{y} \,|\, \boldsymbol{x}] \sum_{j=1}^{m} y_j \widehat{y}_j d_{\operatorname{rank}(j)} \\
&= \sum_{j=1}^{m} \sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{P}[\boldsymbol{s}^j \,|\, y_j, \boldsymbol{x}] \mathbb{P}[y_j \,|\, \boldsymbol{x}] y_j \widehat{y}_j d_{\operatorname{rank}(j)} \\
&= \sum_{j=1}^{m} \sum_{y_j \in \{0,1\}} \mathbb{P}[y_j \,|\, \boldsymbol{x}] y_j \widehat{y}_j d_{\operatorname{rank}(j)} \sum_{\boldsymbol{s}^j \in \mathcal{S}} \mathbb{P}[\boldsymbol{s}^j \,|\, y_j, \boldsymbol{x}] \\
&= \sum_{j=1}^{m} \sum_{y_j \in \{0,1\}} \mathbb{P}[y_j \,|\, \boldsymbol{x}] y_j \widehat{y}_j d_{\operatorname{rank}(j)} \\
&= \sum_{j=1}^{m} \mathbb{P}[y_j = 1 \,|\, \boldsymbol{x}] y_j \widehat{y}_j d_{\operatorname{rank}(j)} = \sum_{j=1}^{m} \eta_j(\boldsymbol{x}) \widehat{y}_j d_{\operatorname{rank}(j)}
\end{aligned} \tag{A.22}$$

Because $\breve{\eta}_j(\boldsymbol{x}) \coloneqq \mathbb{P}\left[\breve{y}_j = 1 \,|\, \boldsymbol{x}\right]$ and $\eta_j(\boldsymbol{x}) = \breve{\eta}_j(\boldsymbol{x})/p_j(\boldsymbol{x})$ (2.26), we get:

$$\begin{aligned}
\Phi_{\mathbb{P}[\boldsymbol{y}\,|\,\boldsymbol{x}]}(\boldsymbol{h}\,|\,\boldsymbol{x}) &= \sum_{j=1}^{m} \frac{\breve{\eta}_j(\boldsymbol{x})\widehat{y}_j d_{\mathrm{rank}(j)}}{p_j(\boldsymbol{x})} \left(u^{j,(1)}(\widehat{y}_j) - u^{j,(0)}(\widehat{y}_j)\right) + u^{j,(0)}(\widehat{y}_j) \\
&= \sum_{j=1}^{m} \frac{\mathbb{P}[\breve{y}_j = 1|\boldsymbol{x}]\breve{y}_j\widehat{y}_j d_{\mathrm{rank}(j)}}{p_j(\boldsymbol{x})} \\
&= \mathbb{E}_{\breve{\boldsymbol{y}} \sim \mathbb{P}[\breve{\boldsymbol{y}}\,|\,\boldsymbol{x}]} \left[\sum_{j=1}^{m} \frac{\breve{y}_j\widehat{y}_j d_{\mathrm{rank}(j)}}{p_j(\boldsymbol{x})} p_j(\boldsymbol{x})\right] \\
&= \mathbb{E}_{\breve{\boldsymbol{y}} \sim \mathbb{P}[\breve{\boldsymbol{y}}\,|\,\boldsymbol{x}]} \left[\widetilde{\mathrm{DCG}}@k(\boldsymbol{x}, \breve{\boldsymbol{y}}, \boldsymbol{h}^{@k}(\boldsymbol{x}))\right] = \breve{\Phi}_{\mathbb{P}[\breve{\boldsymbol{y}}\,|\,\boldsymbol{x}]}(\boldsymbol{h}\,|\,\boldsymbol{x})\,. \quad \text{(A.23)}
\end{aligned}$$

□

A.3 Chapter 6

A.3.1 Order-invariant label-wise utilities as confusion-matrix metrics

In this section, we provide a closer look at the family Ψ_{OI} of order-invariant task utilities that can be decomposed into a sum of label-specific functions as laid out in (5.1) in the main text.

We first formalize the notion of instance-order invariance and then show that this implies the possibility of reformulating any utility function $\Psi(\mathbf{Y}, \widehat{\mathbf{Y}})$ that is invariant under instance reordering in terms of confusion matrices. More precisely, let $\sigma \in \mathfrak{P}(n)$ be a permutation of rows, that is, for $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]^\intercal$, we define $\sigma\mathbf{Y} = [\boldsymbol{y}_{\sigma(1)}, \ldots, \boldsymbol{y}_{\sigma(n)}]^\intercal$. Then we can make the following definition:

Definition A.3.1 (Invariant under instance reordering)**.** Let $n, m \in \mathbb{N}$ and $\Psi : \{0,1\}^{n\times m} \times \{0,1\}^{n\times m} \longrightarrow \mathbb{R}$ be a function of discrete labels and predictions. If Ψ remains unchanged for every permutation of rows $\sigma \in \mathfrak{P}(n)$, i.e.,

$$\Psi(\mathbf{Y}, \widehat{\mathbf{Y}}) = \Psi(\sigma\mathbf{Y}, \sigma\widehat{\mathbf{Y}})\,, \quad \text{(A.24)}$$

then we call Ψ a function that is invariant under instance reordering. We denote the set of instance-order-invariant utilities with m labels as

$$\mathcal{I}_m \coloneqq \left\{\Psi : \{0,1\}^{n\times m} \times \{0,1\}^{n\times m} \longrightarrow \mathbb{R} \colon\ \Psi \text{ is invariant under instance reordering}\right\}. \quad \text{(A.25)}$$

We further define the set of all possible binary confusion matrices with n instances as

$$\mathcal{C}(n) \coloneqq \left\{\widehat{\boldsymbol{c}} \colon n\widehat{\boldsymbol{c}} \in \mathbb{N}^4, \|\widehat{\boldsymbol{c}}\|_{1,1} = 1\right\}, \quad \text{(A.26)}$$

where $\mathbb{N}$ is the set of natural numbers including 0. Now we are ready to provide a lemma for the binary case:

Lemma A.3.2. *Let $\psi_{\mathrm{OI}} : \{0,1\}^n \times \{0,1\}^n \longrightarrow \mathbb{R}$ be a binary loss function that is invariant under instances reordering. Then there exists a function $\psi_{\mathrm{CM}} : \mathcal{C}(n) \longrightarrow \mathbb{R}$ such that $\psi_{\mathrm{OI}} = \psi_{\mathrm{CM}} \circ \widehat{\boldsymbol{c}}$.*

Proof. We provide an explicit construction of ψ_{OI}. To that end, let $\tilde{\widehat{\boldsymbol{c}}} \in \mathcal{C}(n)$ be one such confusion matrix, then there exists $(\boldsymbol{y}, \widehat{\boldsymbol{y}}) \in \{0,1\}^n \times \{0,1\}^n$ given by

$$(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = [\underbrace{(1,1),\dots,(1,1)}_{\times n\cdot\tilde{\widehat{c}}_{\mathrm{tp}}}, \underbrace{(0,1),\dots,(0,1)}_{\times n\cdot\tilde{\widehat{c}}_{\mathrm{fp}}}, \underbrace{(1,0),\dots,(1,0)}_{\times n\cdot\tilde{\widehat{c}}_{\mathrm{fn}}}, \underbrace{(0,0),\dots,(0,0)}_{\times n\cdot\tilde{\widehat{c}}_{\mathrm{tn}}}]^\intercal . \tag{A.27}$$

Define ψ_{CM} such that $\psi_{\mathrm{CM}}(\tilde{\widehat{\boldsymbol{c}}}) = \psi_{\mathrm{OI}}(\boldsymbol{y}, \widehat{\boldsymbol{y}})$. Now, let $(\boldsymbol{y}', \widehat{\boldsymbol{y}}') \in \{0,1\}^n \times \{0,1\}^n$ be an arbitrary label-prediction combination, with $\widehat{\boldsymbol{c}}(\boldsymbol{y}', \widehat{\boldsymbol{y}}') = \tilde{\widehat{\boldsymbol{c}}}$. Then there exists a permutation σ such that $\sigma\boldsymbol{y}' = \boldsymbol{y}$ and $\sigma\widehat{\boldsymbol{y}}' = \widehat{\boldsymbol{y}}$. By the invariance assumption, it holds that

$$\psi_{\mathrm{OI}}(\boldsymbol{y}', \widehat{\boldsymbol{y}}') = \psi_{\mathrm{OI}}(\sigma\boldsymbol{y}', \sigma\widehat{\boldsymbol{y}}') = \psi_{\mathrm{OI}}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = \psi_{\mathrm{CM}}(\tilde{\widehat{\boldsymbol{c}}}) = \psi_{\mathrm{CM}}(\widehat{\boldsymbol{c}}(\boldsymbol{y}', \widehat{\boldsymbol{y}}')). \tag{A.28}$$

As the original $\tilde{\widehat{\boldsymbol{c}}} \in \mathcal{C}(n)$ was arbitrary, the statement is shown. □

We can now extend this lemma to show the equivalence of the two definitions of the task utilities considered in this work:

Theorem A.3.3 (Equivalence of order-invariance and confusion-matrix utilities)**.** *Let $n, m \in \mathbb{N}$, and $\mathcal{Y} = \{0,1\}^m$. Define the set of instance-order invariant, label-averaged utilities as*

$$\mathcal{U}_{\mathrm{OI}} \coloneqq \left\{ \Psi_{\mathrm{OI}} \in \mathcal{I}_n : (\mathbf{Y}, \widehat{\mathbf{Y}}) \mapsto f\left(\psi^1_{\mathrm{OI}}(\boldsymbol{y}_{:,1}, \widehat{\boldsymbol{y}}_{:,1}), \dots, \psi^m_{\mathrm{OI}}(\boldsymbol{y}_{:,m}, \widehat{\boldsymbol{y}}_{:,m})\right) \right\} , \tag{A.29}$$

and the set of confusion-matrix-based, label-averaged utilities as

$$\mathcal{U}_{\mathrm{CM}} \coloneqq \left\{ \Psi_{\mathrm{CM}} : \mathcal{C}(n)^m \longrightarrow \mathbb{R} \text{ s.t. } \widehat{\mathbf{C}} \mapsto f\left(\psi^1_{\mathrm{CM}}(\widehat{\boldsymbol{c}}_1), \dots, \psi^m_{\mathrm{CM}}(\widehat{\boldsymbol{c}}_m)\right) \right\} , \tag{A.30}$$

where f is any aggregation function. Then, these two descriptions are equivalent in the sense that

$$\mathcal{U}_{\mathrm{OI}} = \left\{ \Psi_{\mathrm{CM}} \circ \widehat{\mathbf{C}} \colon \Psi_{\mathrm{CM}} \in \mathcal{U}_{\mathrm{CM}} \right\} , \tag{A.31}$$

that is, every instance-order invariant loss can be written as a confusion-matrix loss, and every confusion-matrix loss leads to an instance-order invariant loss.

Proof. As calculating the confusion matrix is in itself an operation that is invariant under instance reordering, each $\Psi_{\mathrm{CM}} \circ \widehat{\mathbf{C}}$ clearly is instance-order invariant. On the other hand, let $\Psi(\mathbf{Y}, \widehat{\mathbf{Y}}) = \sum_{j=1}^m \psi^j(\boldsymbol{y}_{:,j}, \widehat{\boldsymbol{y}}_{:,j})$, then by Lemma A.3.2 there exist $\psi^1_{\mathrm{CM}}, \dots, \psi^m_{\mathrm{CM}}$ such that

$$\psi^j_{\mathrm{OI}}(\boldsymbol{y}_{:,j}, \widehat{\boldsymbol{y}}_{:,j}) = \psi^j_{\mathrm{CM}}(\widehat{\boldsymbol{c}}(\boldsymbol{y}_{:,j}, \widehat{\boldsymbol{y}}_{:,j})). \tag{A.32}$$

Summing up the individual ψ^j_{CM} gives Ψ_{CM} of the correct form. □

A.3.2 cp-Lipschitz utility functions

Before we prove Theorem 6.3.2 and Theorem 6.6.1, let us first discuss the cp-Lipschitzness condition of the metrics of interest. First, recall the definition from the main text:

Definition 6.3.1 (cp-Lipschitz [Dembczyński et al., 2017])**.** A binary classification metric $\psi(\mathrm{tp}, \mathrm{pp}, \mathrm{cp})$ is said to be cp-Lipschitz if

$$|\psi(\mathrm{tp}, \mathrm{pp}, \mathrm{cp}) - \psi(\mathrm{tp}', \mathrm{pp}', \mathrm{cp}')| \leqslant L_{\mathrm{tp}}(\mathrm{cp})|\mathrm{tp} - \mathrm{tp}'| + L_{\mathrm{pp}}(\mathrm{cp})|\mathrm{pp} - \mathrm{pp}'| + L_{\mathrm{cp}}(\mathrm{cp})|\mathrm{cp} - \mathrm{cp}'|, \tag{6.9}$$

for any $\mathrm{pp}, \mathrm{pp}' \in [0, 1]$, $\mathrm{cp}, \mathrm{cp}' \in (0, 1)$, $0 \leqslant \mathrm{tp} \leqslant \min(\mathrm{cp}, \mathrm{pp})$, and $0 \leqslant \mathrm{tp}' \leqslant \min(\mathrm{cp}', \mathrm{pp}')$. The constants $L_{\mathrm{tp}}(\mathrm{cp}), L_{\mathrm{pp}}(\mathrm{cp}), L_{\mathrm{cp}}(\mathrm{cp})$ are allowed to depend on cp, in contrast to the standard Lipschitz functions.

The rationale behind this definition is that while we need to control the change in the value of the metric under small changes, many popular metrics do not satisfy a standard definition of Lipschitzness (with a global constant). For the same reason, we only require stability for non-trivial problems, that is, in cases where the rate of positives cp is neither zero nor one. Relaxing the definition of Lipschitzness to allow the constants to vary as a function of cp is enough to prove our stability results and regret bounds, while at the same time, it makes more metrics of interest to satisfy the condition, as shown below.

Lemma A.3.4. *The linear confusion-matrix measures defined by* (5.2)*:*

$$\Psi(\widehat{\mathbf{C}}(\mathbf{Y}, \widehat{\mathbf{Y}})) = \sum_{j=1}^{m} \left(g_{j,\mathrm{tn}}\widehat{c}_{j,\mathrm{tn}} + g_{j,\mathrm{fp}}\widehat{c}_{j,\mathrm{fp}} + g_{j,\mathrm{fn}}\widehat{c}_{j,\mathrm{fn}} + g_{j,\mathrm{tp}}\widehat{c}_{j,\mathrm{tp}}\right) \tag{A.33}$$

with fixed coefficient matrix $\mathbf{G}$ *of size* $m \times 4$*, are decomposable functions with* cp*-Lipschitz components.*

Proof. The metric can be rewritten in a decomposable form $\Psi = \sum_j \psi^j$, where each ψ^j in the $(\mathrm{tp}, \mathrm{pp}, \mathrm{cp})$-parameterization has the following form:

$$\psi^j(\mathrm{tp}, \mathrm{pp}, \mathrm{cp}) = T_j \cdot \mathrm{tp} + Q_j \cdot \mathrm{pp} + P_j \cdot \mathrm{cp} + C_j \tag{A.34}$$

where T_j, Q_j, P_j, C_j are some combinations of the coefficient in row j of matrix $\mathbf{G}$. Being a linear function of $\mathrm{tp}, \mathrm{pp}, \mathrm{cp}$, ψ^j is Lipschitz. □

In proposition 1 of Dembczyński et al. [2017] cp-Lipschitzness is proved for binary accuracy, balanced accuracy, F_β-measure, Jaccard, and G-mean. To complement that results, we additionally prove cp-Lipschitzness condition for

recall:

$$|\psi(\mathrm{tp},\mathrm{pp},\mathrm{cp})-\psi(\mathrm{tp}',\mathrm{pp}',\mathrm{cp}')| = \left|\frac{\mathrm{tp}}{\mathrm{cp}}-\frac{\mathrm{tp}'}{\mathrm{cp}'}\right| = \left|\frac{\mathrm{tp}-\mathrm{tp}'}{\mathrm{cp}}+\frac{\mathrm{tp}'}{\mathrm{cp}'}\frac{\mathrm{cp}'-\mathrm{cp}}{\mathrm{cp}}\right|$$
$$\leqslant \underbrace{\frac{1}{\mathrm{cp}}}_{=L_{\mathrm{tp}}(\mathrm{cp})}\left|\mathrm{tp}-\mathrm{tp}'\right| + \underbrace{\frac{\mathrm{tp}'}{\mathrm{cp}'}}_{\leqslant 1}\cdot\underbrace{\frac{1}{\mathrm{cp}}}_{=L_{\mathrm{cp}}(\mathrm{cp})}|\mathrm{cp}-\mathrm{cp}'|. \tag{A.35}$$

The notable exception of metric that is not cp-Lipschitz is precision, due to its behavior for $\mathrm{pp}\to 0$.

A.3.3 Stability of the semi-ETU approximation

We are now ready to prove that when the metric of interest has cp-Lipschitz components, the semi-ETU approximation $\tilde{\Phi}_{\mathrm{ETU}}$ presented in Section 6.3 remains close to the true objective Φ_{ETU}. For the sake of convenience, let us recall the definition of the ETU objective:

$$\Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\mathbf{X}) = \mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\Psi(\widehat{\mathbf{C}}(\mathbf{Y},\widehat{\mathbf{Y}}))\right]$$
$$= \sum_{j=1}^{m}\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\psi^j(\widehat{c}_{j,\mathrm{tp}},\widehat{c}_{j,\mathrm{pp}},\widehat{c}_{j,\mathrm{cp}})\right], \tag{A.36}$$

as well as its approximation:

$$\tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\mathbf{X}) = \sum_{j=1}^{m}\psi^j(\underbrace{\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\widehat{c}_{j,\mathrm{tp}}\right]}_{\tilde{c}_{j,\mathrm{tp}}},\widehat{c}_{j,\mathrm{pp}},\underbrace{\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\widehat{c}_{j,\mathrm{cp}}\right]}_{\tilde{c}_{j,\mathrm{cp}}}). \tag{A.37}$$

We prove the following result:

Theorem 6.3.2. *Let each ψ^j be* cp*-Lipschitz with constants $L^j_{\mathrm{tp}}(\mathrm{cp})$, $L^j_{\mathrm{pp}}(\mathrm{cp})$, $L^j_{\mathrm{cp}}(\mathrm{cp})$. For any $\widehat{\mathbf{Y}}$ it holds:*

$$\left|\Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\mathbf{X})-\tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\mathbf{X})\right| \leqslant \frac{1}{2\sqrt{n}}\left(\sum_{j=1}^{m}\left(L^j_{\mathrm{tp}}(\tilde{c}_{j,\mathrm{cp}})+L^j_{\mathrm{cp}}(\tilde{c}_{j,\mathrm{cp}})\right)\right). \tag{6.10}$$

Proof. For the sake of the analysis, denote the Lipschitz constants as $L^j_{\mathrm{tp}} \coloneqq L^j_{\mathrm{tp}}(\tilde{c}_{j,\mathrm{cp}})$ and $L^j_{\mathrm{cp}} \coloneqq L^j_{\mathrm{cp}}(\tilde{c}_{j,\mathrm{cp}})$. Using definitions (A.36) and (A.37) and applying Jensen's inequality, we have

$$\left|\Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\mathbf{X})-\tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\mathbf{X})\right|$$
$$= \left|\sum_{j=1}^{m}\left(\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\psi^j(\widehat{c}_{j,\mathrm{tp}},\widehat{c}_{j,\mathrm{pp}},\widehat{c}_{j,\mathrm{cp}})\right]-\psi^j(\tilde{c}_{j,\mathrm{tp}},\widehat{c}_{j,\mathrm{pp}},\tilde{c}_{j,\mathrm{cp}})\right)\right|$$
$$\leqslant \sum_{j=1}^{m}\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\left|\psi^j(\widehat{c}_{j,\mathrm{tp}},\widehat{c}_{j,\mathrm{pp}},\widehat{c}_{j,\mathrm{cp}})-\psi^j(\tilde{c}_{j,\mathrm{tp}},\widehat{c}_{j,\mathrm{pp}},\tilde{c}_{j,\mathrm{cp}})\right|\right]. \tag{A.38}$$

We now bound each term in the sum by $(L^j_{\mathrm{tp}} + L^j_{\mathrm{cp}})/(2\sqrt{n})$, which will prove the theorem. For each $j \in [m]$, using cp-Lipschitzness of ψ^j we have:

$$\begin{aligned}
&\left|\psi^j(\widehat{c}_{j,\mathrm{tp}}, \widehat{c}_{j,\mathrm{pp}}, \widehat{c}_{j,\mathrm{cp}}) - \psi^j(\tilde{c}_{j,\mathrm{tp}}, \widehat{c}_{j,\mathrm{pp}}, \tilde{c}_{j,\mathrm{cp}})\right| \\
&\quad = \left|\psi^j(\tilde{c}_{j,\mathrm{tp}}, \widehat{c}_{j,\mathrm{pp}}, \tilde{c}_{j,\mathrm{cp}}) - \psi^j(\widehat{c}_{j,\mathrm{tp}}, \widehat{c}_{j,\mathrm{pp}}, \widehat{c}_{j,\mathrm{cp}})\right| \\
&\quad \leqslant L^j_{\mathrm{tp}}|\widehat{c}_{j,\mathrm{tp}} - \tilde{c}_{j,\mathrm{tp}}| + L^j_{\mathrm{cp}}|\widehat{c}_{j,\mathrm{cp}} - \tilde{c}_{j,\mathrm{cp}}|\,. \qquad \text{(A.39)}
\end{aligned}$$

Taking expectation on both sides gives:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\left|\psi^j(\widehat{c}_{j,\mathrm{tp}}, \widehat{c}_{j,\mathrm{pp}}, \widehat{c}_{j,\mathrm{cp}}) - \psi^j(\tilde{c}_{j,\mathrm{tp}}, \widehat{c}_{j,\mathrm{pp}}, \tilde{c}_{j,\mathrm{cp}})\right|\right] \\
&\quad \leqslant L^j_{\mathrm{tp}}\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[|\widehat{c}_{j,\mathrm{tp}} - \tilde{c}_{j,\mathrm{tp}}|\right] + L^j_{\mathrm{cp}}\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[|\widehat{c}_{j,\mathrm{cp}} - \tilde{c}_{j,\mathrm{cp}}|\right] \\
&\quad = L^j_{\mathrm{tp}}\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\sqrt{(\widehat{c}_{j,\mathrm{tp}} - \tilde{c}_{j,\mathrm{tp}})^2}\right] + L^j_{\mathrm{cp}}\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\sqrt{(\widehat{c}_{j,\mathrm{cp}} - \tilde{c}_{j,\mathrm{cp}})^2}\right] \\
&\quad \leqslant L^j_{\mathrm{tp}}\sqrt{\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[(\widehat{c}_{j,\mathrm{tp}} - \tilde{c}_{j,\mathrm{tp}})^2\right]} \\
&\quad + L^j_{\mathrm{cp}}\sqrt{\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[(\widehat{c}_{j,\mathrm{cp}} - \tilde{c}_{j,\mathrm{cp}})^2\right]}\,, \quad (6.13) \qquad \text{(A.40)}
\end{aligned}$$

where the last inequality follows from Jensen's inequality applied to a concave function $x \mapsto \sqrt{x}$. Using the fact that $\tilde{c}_{j,\mathrm{tp}} = \mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}[\widehat{c}_{j,\mathrm{tp}}]$, we have

$$\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[(\widehat{c}_{j,\mathrm{tp}} - \tilde{c}_{j,\mathrm{tp}})^2\right] = \mathrm{Var}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}(\widehat{c}_{j,\mathrm{tp}}) \leqslant \frac{1}{4n}\,, \qquad \text{(A.41)}$$

as $\widehat{c}_{j,\mathrm{tp}} = \frac{1}{n}\sum_{i=1}^n y_{i,j}\widehat{y}_{i,j}$ and is an average of n Bernoulli i.i.d. random variables $y_{i,j}\widehat{y}_{i,j}$ each having variance at most $\frac{1}{4}$. The same reasoning can be apply to to $\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}[(\widehat{c}_{j,\mathrm{cp}} - \tilde{c}_{j,\mathrm{cp}})^2] \leqslant \frac{1}{4n}$. By applying these to (6.13) we get:

$$\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}[\mathbf{Y}\,|\,\mathbf{X}]}\left[\left|\psi^j(\tilde{c}_{j,\mathrm{tp}}, \widehat{c}_{j,\mathrm{pp}}, \tilde{c}_{j,\mathrm{cp}}) - \psi^j(\widehat{c}_{j,\mathrm{tp}}, \widehat{c}_{j,\mathrm{pp}}, \widehat{c}_{j,\mathrm{cp}})\right|\right] \leqslant \frac{L^j_{\mathrm{tp}} + L^j_{\mathrm{cp}}}{2\sqrt{n}} \qquad \text{(A.42)}$$

what finishes the proof. □

A.3.4 Regret of semi-ETU under model misspecification

In this section, we quantify the influence of the estimation error of marginal probabilities, proving Theorem 6.6.1. To emphasize the dependence of Φ_{ETU} on the label marginal probability estimates, in this section we will write

$$\Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \mathbf{X}) = \mathbb{E}_{\mathbf{Y}\sim\boldsymbol{\eta}(\mathbf{X})}\Psi(\mathbf{Y}, \widehat{\mathbf{Y}}) =: \Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \boldsymbol{\eta})\,, \qquad \text{(A.43)}$$

This notation is well-defined, as we have shown in Section 6.2 that in fact, the dependence on $\mathbf{X}$ is mediated only through the marginal label probabilities $\boldsymbol{\eta}(\mathbf{X}) =$

$[\boldsymbol{\eta}(\boldsymbol{x}_1), \dots, \boldsymbol{\eta}(\boldsymbol{x}_n)]$, and we abbreviate $\boldsymbol{\eta}(\mathbf{X})$ as $\boldsymbol{\eta}$. Similarly, we will write

$$\tilde{c}_{j,\mathrm{tp}}(\boldsymbol{\eta}) = \mathbb{E}_{\boldsymbol{y}_{:,j} \sim \eta_j(\mathbf{X})}[\mathrm{tp}(\boldsymbol{y}_{:,j}, \widehat{\boldsymbol{y}}_{:,j})] = \frac{1}{m} \sum_{i=1}^{n} \eta_j(\boldsymbol{x}_i) \widehat{y}_{i,j} \,,$$
$$\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta}) = \mathbb{E}_{\boldsymbol{y}_{:,j} \sim \eta_j(\mathbf{X})}[\mathrm{cp}(\boldsymbol{y}_{:,j})] = \frac{1}{m} \sum_{i=1}^{n} \eta_j(\boldsymbol{x}_i) \,. \tag{A.44}$$

Note that pp is independent of $\boldsymbol{\eta}$. This allows us to write the semi-ETU objective as

$$\tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \boldsymbol{\eta}) \coloneqq \Psi(\tilde{\boldsymbol{c}}_{:,\mathrm{tp}}, \widehat{\boldsymbol{c}}_{:,\mathrm{pp}}, \tilde{\boldsymbol{c}}_{:,\mathrm{cp}}) \,, \tag{A.45}$$

where we dropped the dependence of $\tilde{\boldsymbol{c}}_{:,\mathrm{tp}}$ and $\tilde{\boldsymbol{c}}_{:,\mathrm{cp}}$ on $\boldsymbol{\eta}$ as clear from the context. The optimal (Bayes) predictor $\mathbf{Y}^\star$ is the one which maximizes the expected utility with regard to the true label probabilities:

$$\mathbf{Y}^\star \in \operatorname*{arg\,max}_{\widehat{\mathbf{Y}} \in \mathcal{Y}^{@k,n}} \Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \boldsymbol{\eta}) \,. \tag{A.46}$$

Unfortunately, the learning algorithm does not know the true label marginals $\boldsymbol{\eta}(\mathbf{X})$, but has only access to the estimates $\widehat{\boldsymbol{\eta}}(\mathbf{X}) = [\widehat{\boldsymbol{\eta}}(\boldsymbol{x}_1), \dots, \widehat{\boldsymbol{\eta}}(\boldsymbol{x}_n)]$ for the considered set of instances. The algorithm computes its predictions $\mathbf{Y}^\dagger$ by using the estimates in place of the true marginals. Thus, it can only generate

$$\mathbf{Y}^\dagger \in \operatorname*{arg\,max}_{\widehat{\mathbf{Y}} \in \mathcal{Y}^{@k,n}} \Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \widehat{\boldsymbol{\eta}}) \,. \tag{A.47}$$

We can make the same definitions also for the semi-empirical ETU optimization, leading to

$$\tilde{\mathbf{Y}}^\star \in \operatorname*{arg\,max}_{\widehat{\mathbf{Y}} \in \mathcal{Y}^{@k,n}} \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \boldsymbol{\eta}) \quad \text{and} \quad \tilde{\mathbf{Y}}^\dagger \in \operatorname*{arg\,max}_{\widehat{\mathbf{Y}} \in \mathcal{Y}^{@k,n}} \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \widehat{\boldsymbol{\eta}}) \,. \tag{A.48}$$

Regret for semi-ETU approximation First, we show that for metrics with cp-Lipschitz components, the Ψ-regret of the resulting semi-ETU predictor (A.48), which is the suboptimality of $\tilde{\mathbf{Y}}^\dagger$ (with respect to $\mathbf{Y}^\star$) in terms of Φ_{ETU}, is well-controlled and upper-bounded by the estimation error of the marginals. As the resulting expression is somewhat unwieldy, we then apply some further bounding to arrive at the much simpler result stated in the main paper in Theorem 6.6.1.

Lemma A.3.5 (Misspecification for semi-ETU approximation)**.** *Let $\Psi \in \mathcal{U}_{\mathrm{OI}}$ be an instance-order invariant linearly decomposable loss function that is* cp*-Lipschitz, and $\widehat{\mathbf{Y}}$ an arbitrary set of predictions. Then, the difference of the semi-empirical ETU risk when using two different versions of the marginals, $\boldsymbol{\eta}$ and $\boldsymbol{\eta}'$, is bounded by the difference $\boldsymbol{\eta} - \boldsymbol{\eta}'$ through*

$$\begin{aligned} &|\tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \boldsymbol{\eta}) - \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \boldsymbol{\eta}')| \\ &\qquad \leqslant \frac{1}{n} \sum_{j=1}^{m} \Big(L^j_{\mathrm{tp}}(\tilde{c}_{j,\mathrm{tp}}(\boldsymbol{\eta})) + L^j_{\mathrm{cp}}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) \Big) \sum_{i=1}^{n} \left| \eta_j(\boldsymbol{x}_i) - \eta'_j(\boldsymbol{x}_i) \right| \,. \end{aligned} \tag{A.49}$$

Proof. Plugging in the definitions, we have

$$\begin{aligned}
&|\tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\boldsymbol{\eta}) - \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\boldsymbol{\eta}')| \\
&\quad= \left|\sum_{j=1}^{m} \psi^j(\tilde{c}_{j,\mathrm{tp}}(\boldsymbol{\eta}), \widehat{c}_{j,\mathrm{pp}}, \tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) - \sum_{j=1}^{m} \psi^j(\tilde{c}_{j,\mathrm{tp}}(\boldsymbol{\eta}'), \widehat{c}_{j,\mathrm{pp}}, \tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta}'))\right| \\
&\quad\leqslant \sum_{j=1}^{m} \left|\psi^j(\tilde{c}_{j,\mathrm{tp}}(\boldsymbol{\eta}), \widehat{c}_{j,\mathrm{pp}}, \tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) - \psi^j(\tilde{c}_{j,\mathrm{tp}}(\boldsymbol{\eta}'), \widehat{c}_{j,\mathrm{pp}}, \tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta}'))\right| \\
&\quad\leqslant \sum_{j=1}^{m} L^j_{\mathrm{tp}}(\tilde{c}_{j,\mathrm{tp}}(\boldsymbol{\eta})) \left|\tilde{c}_{j,\mathrm{tp}}(\boldsymbol{\eta}) - \tilde{c}_{j,\mathrm{tp}}(\boldsymbol{\eta}')\right| + L^j_{\mathrm{cp}}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) \left|\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta}) - \tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta}')\right| ,
\end{aligned} \tag{A.50}$$

where the last line used the definition of cp-Lipschitzness. The terms under the sum can be bounded by the estimation error of $\boldsymbol{\eta}'$:

$$\begin{aligned}
\left|\tilde{c}_{j,\mathrm{tp}}(\boldsymbol{\eta}) - \tilde{c}_{j,\mathrm{tp}}(\boldsymbol{\eta}')\right| &= \left|\frac{1}{n}\sum_{i=1}^{n} \widehat{y}_{i,j}(\eta_j(\boldsymbol{x}_i) - \eta'_j(\boldsymbol{x}_i))\right| \leqslant \frac{1}{n}\sum_{i=1}^{n} \left|\eta_j(\boldsymbol{x}_i) - \eta'_j(\boldsymbol{x}_i)\right| , \\
\left|\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta}) - \tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta}')\right| &= \left|\frac{1}{n}\sum_{i=1}^{n} (\eta_j(\boldsymbol{x}_i) - \eta'_j(\boldsymbol{x}_i))\right| \leqslant \frac{1}{n}\sum_{i=1}^{n} \left|\eta_j(\boldsymbol{x}_i) - \eta'_j(\boldsymbol{x}_i)\right| ,
\end{aligned} \tag{A.51}$$

where in the first line we used $\widehat{y}_{i,j} \in \{0,1\}$. □

Lemma A.3.6 (Regret bound for semi-ETU approximation)**.** *Let $\Psi \in \mathcal{U}_{\mathrm{OI}}$ be an instance-order invariant linearly decomposable loss function that is* cp*-Lipschitz. Then we have*

$$\begin{aligned}
&\Phi_{\mathrm{ETU}}(\mathbf{Y}^\star;\mathbf{X}) - \Phi_{\mathrm{ETU}}(\mathbf{Y}^\dagger;\mathbf{X}) \\
&\quad\leqslant \frac{1}{\sqrt{n}} \sum_{j=1}^{m} \left(L^j_{\mathrm{tp}}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) + L^j_{\mathrm{cp}}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta}))\right) \\
&\qquad + 2\sum_{j=1}^{m} \frac{L^j_{\mathrm{tp}}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) + L^j_{\mathrm{cp}}(\widehat{c}_{j,\mathrm{cp}}(\boldsymbol{\eta}))}{n} \sum_{i=1}^{n} |\eta_j(\boldsymbol{x}_i) - \widehat{\eta}_j(\boldsymbol{x}_i)| .
\end{aligned} \tag{A.52}$$

Proof. Using the optimality of $\mathbf{Y}^\dagger$ and a supremum bound, we obtain

$$\begin{aligned}
\Phi_{\mathrm{ETU}}(\mathbf{Y}^\star;\boldsymbol{\eta}) - \Phi_{\mathrm{ETU}}(\mathbf{Y}^\dagger;\boldsymbol{\eta}) &= \Phi_{\mathrm{ETU}}(\mathbf{Y}^\star;\boldsymbol{\eta}) - \tilde{\Phi}_{\mathrm{ETU}}(\mathbf{Y}^\star;\widehat{\boldsymbol{\eta}}) \\
&\quad + \tilde{\Phi}_{\mathrm{ETU}}(\mathbf{Y}^\star;\widehat{\boldsymbol{\eta}}) - \tilde{\Phi}_{\mathrm{ETU}}(\mathbf{Y}^\dagger;\widehat{\boldsymbol{\eta}}) \\
&\quad + \tilde{\Phi}_{\mathrm{ETU}}(\mathbf{Y}^\dagger;\widehat{\boldsymbol{\eta}}) - \Phi_{\mathrm{ETU}}(\mathbf{Y}^\dagger;\boldsymbol{\eta}) \\
&\leqslant \Phi_{\mathrm{ETU}}(\mathbf{Y}^\star;\boldsymbol{\eta}) - \tilde{\Phi}_{\mathrm{ETU}}(\mathbf{Y}^\star;\widehat{\boldsymbol{\eta}}) \\
&\quad + \tilde{\Phi}_{\mathrm{ETU}}(\mathbf{Y}^\dagger;\widehat{\boldsymbol{\eta}}) - \Phi_{\mathrm{ETU}}(\mathbf{Y}^\dagger;\boldsymbol{\eta}) \\
&\leqslant 2\sup_{\widehat{\mathbf{Y}}} \left|\Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\boldsymbol{\eta}) - \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\widehat{\boldsymbol{\eta}})\right| .
\end{aligned} \tag{A.53}$$

We then use Theorem 6.3.2 and Lemma A.3.5 to bound

$$\begin{aligned}
&\left|\Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\boldsymbol{\eta}) - \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\widehat{\boldsymbol{\eta}})\right| \\
&\quad= \left|\Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\boldsymbol{\eta}) - \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\boldsymbol{\eta}) + \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\boldsymbol{\eta}) - \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\widehat{\boldsymbol{\eta}})\right| \\
&\quad\leqslant \left|\Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\boldsymbol{\eta}) - \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\boldsymbol{\eta})\right| + \left|\tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\boldsymbol{\eta}) - \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}};\widehat{\boldsymbol{\eta}})\right| \\
&\quad\leqslant \frac{1}{2\sqrt{n}}\left(\sum_{j=1}^{m}(L^j_{\mathrm{tp}}(\tilde{c}_{j,\mathrm{cp}}) + L^j_{\mathrm{cp}}(\tilde{c}_{j,\mathrm{cp}}))\right) \\
&\qquad+ \sum_{j=1}^{m}\frac{L^j_{\mathrm{tp}}(\tilde{c}_{j,\mathrm{cp}}) + L^j_{\mathrm{cp}}(\tilde{c}_{j,\mathrm{cp}})}{n}\sum_{i=1}^{n}\left|\eta_j(\boldsymbol{x}_i) - \widehat{\eta}_j(\boldsymbol{x}_i)\right|, \qquad \text{(A.54)}
\end{aligned}$$

where we use the shorthand $\tilde{c}_{j,\mathrm{cp}} = \tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})$. □

At the cost of having a less strict bound, we can simplify this to

Theorem 6.6.1. *Let $\tilde{\mathbf{Y}}^\dagger$ be defined as above. Under the assumptions of Theorem 6.3.2:*

$$\begin{aligned}
\mathrm{Reg}_{\mathrm{ETU}}\left(\tilde{\mathbf{Y}}^\dagger;\mathbf{X}\right) &= \Phi_{\mathrm{ETU}}\left(\mathbf{Y}^\star;\mathbf{X}\right) - \Phi_{\mathrm{ETU}}\left(\tilde{\mathbf{Y}}^\dagger;\mathbf{X}\right) \\
&\leqslant \frac{m}{\sqrt{n}}B + 2\frac{\sqrt{m}}{n}B\sum_{i=1}^{n}\left\|\boldsymbol{\eta}(\boldsymbol{x}_i) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x}_i)\right\|_2, \qquad \text{(6.29)}
\end{aligned}$$

where $B \coloneqq \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(L^j_{\mathrm{tp}}(\tilde{c}_{j,\mathrm{cp}}) + L^j_{\mathrm{cp}}(\widehat{c}_{j,\mathrm{cp}})\right)^2}$ is the quadratic mean of the Lipschitz constants.

Proof. Using the Cauchy-Schwarz inequality, we can bound

$$\begin{aligned}
&\sum_{j=1}^{m}\frac{L^j_{\mathrm{tp}}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) + L^j_{\mathrm{cp}}(\widehat{c}_{j,\mathrm{cp}}(\boldsymbol{\eta}))}{n}\sum_{i=1}^{n}\left|\eta_j(\boldsymbol{x}_i) - \widehat{\eta}_j(\boldsymbol{x}_i)\right| \\
&\quad\leqslant \frac{1}{m}\sum_{i=1}^{n}\sqrt{\sum_{j=1}^{m}\left(L^j_{\mathrm{tp}}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) + L^j_{\mathrm{cp}}(\widehat{c}_{j,\mathrm{cp}}(\boldsymbol{\eta}))\right)^2}\cdot\sqrt{\sum_{j=1}^{m}(\eta_j(\boldsymbol{x}_i) - \widehat{\eta}_j(\boldsymbol{x}_i))^2} \\
&\quad= \frac{1}{m}\sqrt{\sum_{j=1}^{m}\left(L^j_{\mathrm{tp}}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) + L^j_{\mathrm{cp}}(\widehat{c}_{j,\mathrm{cp}}(\boldsymbol{\eta}))\right)^2}\cdot\sum_{i=1}^{n}\sqrt{\left\|\boldsymbol{\eta}(\boldsymbol{x}_i) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x}_i)\right\|_2^2} \\
&\quad= \frac{\sqrt{m}}{n}B\sum_{i=1}^{n}\left\|\boldsymbol{\eta}(\boldsymbol{x}_i) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x}_i)\right\|_2. \qquad \text{(A.55)}
\end{aligned}$$

For the other term, we can use the inequality between the arithmetic and quadratic

means, so that

$$\sum_{j=1}^{m} \left(L_{\mathrm{tp}}^{j}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) + L_{\mathrm{cp}}^{j}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) \right) \leqslant m \sqrt{\frac{1}{m} \sum_{j=1}^{m} \left(L_{\mathrm{tp}}^{j}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) + L_{\mathrm{cp}}^{j}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) \right)^2} = mB\,. \tag{A.56}$$

□

A.3.5 Regret for non-approximated ETU

We can formulate the equivalent of Lemma A.3.6 also for the maximizer of the true empirical ETU risk, $\mathbf{Y}^{\dagger}$:

Lemma A.3.7 (Regret bound for ETU maximization)**.** *Let $\Psi \in \mathcal{U}_{\mathrm{OI}}$ be an instance-order invariant linearly decomposable loss function that is* cp*-Lipschitz. Then we have*

$$\begin{aligned}
&\Phi_{\mathrm{ETU}}(\mathbf{Y}^{\star}; \mathbf{X}) - \Phi_{\mathrm{ETU}}(\mathbf{Y}^{\dagger}; \mathbf{X}) \\
&\quad \leqslant \frac{1}{\sqrt{n}} \sum_{j=1}^{m} \left(L_{\mathrm{tp}}^{j}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) + L_{\mathrm{tp}}^{j}(\tilde{c}_{j,\mathrm{cp}}(\widehat{\boldsymbol{\eta}})) + L_{\mathrm{cp}}^{j}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) + L_{\mathrm{cp}}^{j}(\tilde{c}_{j,\mathrm{cp}}(\widehat{\boldsymbol{\eta}})) \right) \\
&\qquad + 2 \sum_{j=1}^{m} \frac{L_{\mathrm{tp}}^{j}(\tilde{c}_{j,\mathrm{cp}}(\boldsymbol{\eta})) + L_{\mathrm{cp}}^{j}(\widehat{c}_{j,\mathrm{cp}}(\boldsymbol{\eta}))}{n} \sum_{i=1}^{n} \left| \eta_j(\boldsymbol{x}_i) - \widehat{\eta}_j(\boldsymbol{x}_i) \right|\,.
\end{aligned} \tag{A.57}$$

Proof. Following the same line of argument as for Lemma A.3.6, we get

$$\begin{aligned}
\Phi_{\mathrm{ETU}}(\mathbf{Y}^{\star}; \boldsymbol{\eta}) - \Phi_{\mathrm{ETU}}(\mathbf{Y}^{\dagger}; \boldsymbol{\eta}) &= \Phi_{\mathrm{ETU}}(\mathbf{Y}^{\star}; \boldsymbol{\eta}) - \Phi_{\mathrm{ETU}}(\mathbf{Y}^{\star}; \widehat{\boldsymbol{\eta}}) \\
&\quad + \Phi_{\mathrm{ETU}}(\mathbf{Y}^{\star}; \widehat{\boldsymbol{\eta}}) - \Phi_{\mathrm{ETU}}(\mathbf{Y}^{\dagger}; \widehat{\boldsymbol{\eta}}) \\
&\quad + \Phi_{\mathrm{ETU}}(\mathbf{Y}^{\dagger}; \widehat{\boldsymbol{\eta}}) - \Phi_{\mathrm{ETU}}(\mathbf{Y}^{\dagger}; \boldsymbol{\eta}) \\
&\leqslant \Phi_{\mathrm{ETU}}(\mathbf{Y}^{\star}; \boldsymbol{\eta}) - \Phi_{\mathrm{ETU}}(\mathbf{Y}^{\star}; \widehat{\boldsymbol{\eta}}) \\
&\quad + \Phi_{\mathrm{ETU}}(\mathbf{Y}^{\dagger}; \widehat{\boldsymbol{\eta}}) - \Phi_{\mathrm{ETU}}(\mathbf{Y}^{\dagger}; \boldsymbol{\eta}) \\
&\leqslant 2 \sup_{\widehat{\mathbf{Y}}} \left| \Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \boldsymbol{\eta}) - \Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \widehat{\boldsymbol{\eta}}) \right|\,.
\end{aligned} \tag{A.58}$$

Next, we make use of the semi-empirical ETU risk to bound

$$\begin{aligned}
\left| \Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \boldsymbol{\eta}) - \Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \widehat{\boldsymbol{\eta}}) \right| &\leqslant \left| \Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \boldsymbol{\eta}) - \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \boldsymbol{\eta}) \right| \\
&\quad + \left| \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \boldsymbol{\eta}) - \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \widehat{\boldsymbol{\eta}}) \right| \\
&\quad + \left| \tilde{\Phi}_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \widehat{\boldsymbol{\eta}}) - \Phi_{\mathrm{ETU}}(\widehat{\mathbf{Y}}; \widehat{\boldsymbol{\eta}}) \right|
\end{aligned} \tag{A.59}$$

The individual terms can now again be bounded by Theorem 6.3.2 and Lemma A.3.5.

□

A.4 Chapter 7

As in Chapter 7, in this appendix we use $\boldsymbol{h}$ to refer to $\boldsymbol{h}^{\mathrm{rnd}@k}$, as defined in Section 7.1.

A.4.1 Madows sampling

In this section, we present a sampling scheme for the following sampling problem: given a real-valued vector $\boldsymbol{\theta} \in [0,1]^m$ of marginal probabilities with $\|\boldsymbol{\theta}\|_1 = k$, sample binary vectors $\widehat{\boldsymbol{y}} \in \{0,1\}^m$ such that the distribution of $\widehat{\boldsymbol{y}}$ has $\boldsymbol{\theta}$ as the marginals, $\mathbb{E}[\widehat{\boldsymbol{y}}] = \boldsymbol{\theta}$

Theorem A.4.1. *Let $m \geqslant 1$. Given a vector $\boldsymbol{\theta} \in [0,1]^m$ satisfying $\|\boldsymbol{\theta}\|_1 = k$, Algorithm A.1 returns a randomized binary vector $\widehat{\boldsymbol{y}} \in \{0,1\}^m$ of size k, $\|\widehat{\boldsymbol{y}}\|_1 = k$, with marginals given by $\boldsymbol{\theta}$, $\mathbb{E}[\widehat{\boldsymbol{y}}] = \boldsymbol{\theta}$. The algorithm runs in $O(m)$ time.*

Algorithm A.1 Madow's sampling scheme

Require: a vector of marginals $\boldsymbol{\theta} \in [0,1]^m$ with $\|\boldsymbol{\theta}\|_1 = k$
Ensure: a random vector $\widehat{\boldsymbol{y}} \in \{0,1\}^m$ with $\|\widehat{\boldsymbol{y}}\|_1 = k$ such that $\mathbb{E}[\widehat{\boldsymbol{y}}] = \boldsymbol{\theta}$
1: compute $\Theta_0 = 0$ and $\Theta_j = \Theta_{j-1} + \theta_j$ for $j = 1, \ldots, m$
2: sample a uniformly distributed random variable U from the interval $[0,1]$
3: $\widehat{\boldsymbol{y}} = \mathbf{0}$
4: **for** $i \in \{0, 1, \ldots, k-1\}$ **do**
5: select j such that $\Theta_{j-1} < U + i \leqslant \Theta_j$
6: $\widehat{y}_j = 1$
7: **return** $\widehat{\boldsymbol{y}}$

The algorithm is due to Madow [Madow, 1949, Mukhopadhyay et al., 2022], and the considered sampling problem has been studied in the statistical literature under the name unequal probability sampling design [Hanif and Brewer, 1980].

Below, we give a simple proof of correctness of the algorithm for completeness.

Proof. First note that for any $i \in \{0, 1, \ldots, k-1\}$, there exists a unique j for which $\Theta_{j-1} < U + i \leqslant \Theta_j$. This is because due to $\sum_{j=1}^m \theta_j = k$, the intervals $(\Theta_0, \Theta_1], (\Theta_1, \Theta_2], \ldots, (\Theta_{m-1}, \Theta_m]$ are disjoint and cover $(0, k]$, whereas $U + i \in (0, k]$ with probability one. Furthermore, the algorithm will select distinct j's for distinct i's. This is because the condition $\Theta_{j-1} < U + i \leqslant \Theta_j$ is equivalent to $i \in (\Theta_{j-1} - U, \Theta_j - U]$, and the interval $(\Theta_{j-1} - U, \Theta_j - U]$ have width $\theta_j \leqslant 1$ and thus can contain at most one integer. So the algorithm will return $\widehat{\boldsymbol{y}}$ with exactly k ones.

Since $\mathbb{E}[\widehat{y}_j] = \mathbb{P}[\widehat{y}_j = 1]$, we need to show that this probability is equal to θ_j

for each j. We have

$$\mathbb{P}[\widehat{y}_j = 1] = \mathbb{P}\left[\boldsymbol{x}, \boldsymbol{y}\right]\left[U \in \bigcup_{i=0}^{k-1}(\Theta_{j-1} - i, \Theta_j - i]\right]$$
$$= (0,1] \cap \bigcup_{i=0}^{k-1}(\Theta_{j-1} - i, \Theta_j - i] = \Theta_j - \Theta_{j-1} = \theta_j.$$

□

The theorem and the algorithm from its proof can then be used to generate k-hot encoded prediction vectors independently for any instance of interest $\boldsymbol{x}$.

A.4.2 The optimal classifier for linear metrics under PU setting

Theorem 7.2.1 (Regret for linear utilities under PU framework)**.** *The optimal classifier* $\boldsymbol{h}^\star \in \arg\max_{\boldsymbol{h}\in\mathcal{H}^{@k}} \Psi(\boldsymbol{h})$ *for* $\Psi(\boldsymbol{h}) = \sum_{j=1}^m \boldsymbol{g}_j \cdot \boldsymbol{c}_j$ *and any gain matrix* $\mathbf{G}$ *is given by:*

$$\boldsymbol{h}^\star(\boldsymbol{x}) \coloneqq \text{select-top-}k\left(\boldsymbol{a} \odot \boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{b}\right), \tag{7.8}$$

where:

$$\boldsymbol{a} = \boldsymbol{g}_{:,\text{tn}} + \boldsymbol{g}_{:,\text{tp}} - \boldsymbol{g}_{:,\text{fp}} - \boldsymbol{g}_{:,\text{fn}}, \qquad \boldsymbol{b} = \boldsymbol{g}_{:,\text{fp}} - \boldsymbol{g}_{:,\text{tn}}. \tag{7.9}$$

Proof. The linear metric is decomposable over instances as:

$$\begin{aligned}\boldsymbol{g}_j \cdot \boldsymbol{c}_j(\boldsymbol{h}) &= g_{j,\text{tn}}\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}[(1-\eta_j(\boldsymbol{x}))(1-\theta_j(\boldsymbol{x}))] \\ &\quad + g_{j,\text{fp}}\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}[(1-\eta_j(\boldsymbol{x}))\theta_j(\boldsymbol{x})] \\ &\quad + g_{j,\text{fn}}\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}[\eta_j(\boldsymbol{x})(1-\theta_j(\boldsymbol{x}))] \\ &\quad + g_{j,\text{tp}}\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}[\eta_j(\boldsymbol{x})\theta_j(\boldsymbol{x})] \\ &= \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}[(a_j\eta_j(\boldsymbol{x}) + b_j)\theta_j(\boldsymbol{x})] + r_j,\end{aligned} \tag{A.60}$$

where a_j and b_j are as stated in the theorem, while

$$r_j = g_{j,\text{tn}}\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}[1-\eta_j(\boldsymbol{x})] + g_{j,\text{fn}}\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}[\eta_j(\boldsymbol{x})]. \tag{A.61}$$

Thus, we can rewrite the objective as:

$$\Psi(\boldsymbol{h}) = \sum_j \boldsymbol{g}_j \cdot \boldsymbol{c}_j(\boldsymbol{h}) = \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\left[\sum_{j=1}^m (a_j\eta_j(\boldsymbol{x}) + b_j)\theta_j(\boldsymbol{x})\right] + \boldsymbol{r}, \tag{A.62}$$

where $R = \sum_{j=1}^m r_j$ does not depend on $\boldsymbol{h}$. For each $\boldsymbol{x} \in \mathcal{X}$, the objective can be independently maximized by the choice of $\boldsymbol{h}(\boldsymbol{x}) \in \mathcal{H}^{@k}$ which maximizes $\sum_{j=1}^m (a_j\eta_j(\boldsymbol{x}) + b_j)\theta_j(\boldsymbol{x})$. This is achieved by sorting $a_j\eta_j(\boldsymbol{x}) + b_j$ in a descending order, and setting $\theta_j(\boldsymbol{x}) = 1$ for the top k coordinates, and $\theta_j(\boldsymbol{x}) = 0$ for the remaining coordinates (with ties broken arbitrarily), resulting in $\boldsymbol{\theta}(\boldsymbol{x}) = \text{select-top-}k\left(\boldsymbol{a} \odot \boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{b}\right)$. □

A.4.3 The optimal classifier for general metrics under PU setting

In this section, we prove the existence and the form of the optimal classifier. Our results extend past results on binary classification [Koyejo et al., 2014] and multi-class classification [Narasimhan et al., 2015]. We first show that the set of confusion matrices achievable by randomized k-budgeted classifiers $\boldsymbol{h}^{\mathrm{rnd}@k}$ is a compact set. Then, the statement of the main theorem follows from the first-order optimality conditions as well as the absolute continuity of marginal vector $\boldsymbol{\eta}(\boldsymbol{x})$. We stress that the results here are general and applicable to any multi-label utility satisfying Assumption 7.2.3, which need not necessarily be a macro-averaged utility.

We remind that the set of confusion matrices achievable by randomized k-budgeted classifiers on distribution $\mathbb{P}[\boldsymbol{x}, \boldsymbol{y}]$, is denoted as

$$\mathcal{C}_{\mathbb{P}}^{m,@k} = \mathcal{C}_{\mathbb{P}}^{m,@k} = \left\{\mathbf{C}(\boldsymbol{h}) \;:\; \boldsymbol{h} \in \mathcal{H}^{@k}\right\}, \tag{A.63}$$

and that optimizing the metric $\Psi(\boldsymbol{h})$ over $\boldsymbol{h} \in \mathcal{H}^{@k}$ is equivalent to optimizing $\Psi(\mathbf{C})$ over $\mathbf{C} \in \mathcal{C}_{\mathbb{P}}^{m}$.

Lemma A.4.2. *$\mathcal{C}_{\mathbb{P}}^{m,@k}$ is a convex set.*

Proof. Take any $\mathbf{C}_1, \mathbf{C}_2 \in \mathcal{C}_{\mathbb{P}}^{m,@k}$ and any $\lambda \in [0,1]$, and we show that $\mathbf{C}_\lambda = \lambda\mathbf{C}_1 + (1-\lambda)\mathbf{C}_2 \in \mathcal{C}_{\mathbb{P}}^{m,@k}$. Since $\mathbf{C}_1, \mathbf{C}_2 \in \mathcal{C}_{\mathbb{P}}^{m,@k}$, there exist k-budgeted randomized classifiers $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$, such that $\mathbf{C}_1 = \mathbf{C}(\boldsymbol{h}_1)$ and $\mathbf{C}_2 = \mathbf{C}(\boldsymbol{h}_2)$. Take $\boldsymbol{h}_\lambda$ defined on $\boldsymbol{\theta}_\lambda(\boldsymbol{x}) = \lambda\boldsymbol{\theta}_1(\boldsymbol{x}) + (1-\lambda)\boldsymbol{\theta}_2(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathcal{X}$. Since $\Delta^{@k}$ is convex and $\boldsymbol{\theta}_1(\boldsymbol{x}), \boldsymbol{\theta}_2(\boldsymbol{x}) \in \Delta^{@k}$ for all $\boldsymbol{x} \in \mathcal{X}$, it also holds that $\boldsymbol{\theta}_\lambda(\boldsymbol{x}) \in \Delta^{@k}$ for all $\boldsymbol{x} \in \mathcal{X}$, so $\boldsymbol{h}_\lambda$ is also k-budgeted randomized classifier. Since the confusion matrix is linear in predictions, we have $\mathbf{C}(\boldsymbol{h}_\lambda) = \lambda\mathbf{C}(\boldsymbol{h}_1) + (1-\lambda)\mathbf{C}(\boldsymbol{h}_2) = \mathbf{C}_\lambda$, which proves that $\mathbf{C}(\boldsymbol{h}_\lambda) \in \mathcal{C}_{\mathbb{P}}^{m,@k}$. □

We now argue that for the analysis of $\mathcal{C}_{\mathbb{P}}^{m,@k}$, it suffices to consider classifiers with $\boldsymbol{\theta}(\boldsymbol{x})$ of form $\boldsymbol{\theta} = \boldsymbol{f} \circ \boldsymbol{\eta}$, i.e. $\boldsymbol{\theta}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{\eta}(\boldsymbol{x}))$ for some function $\boldsymbol{f}\colon [0,1]^m \to \Delta^{@k}$.

Lemma A.4.3. *For any $\boldsymbol{\theta} \in \mathcal{S}^{@k}$, there exists function $\boldsymbol{f}\colon [0,1]^m \to \Delta^{@k}$ such that $\boldsymbol{\theta}$ and $\boldsymbol{f} \circ \boldsymbol{\eta}$ have the same confusion matrices.*

Proof. If $\boldsymbol{\theta}$ is not of the form $\boldsymbol{f} \circ \boldsymbol{\eta}$, we define function $\boldsymbol{f}$ as:

$$\boldsymbol{f}(\boldsymbol{\eta}) = \mathbb{E}\left[\boldsymbol{\theta}(\boldsymbol{x}) | \boldsymbol{\eta}(\boldsymbol{x}) = \boldsymbol{\eta}\right]. \tag{A.64}$$

Due to convexity of $\Delta^{@k}$, we have $\boldsymbol{f}(\boldsymbol{\eta}) \in \Delta^{@k}$. Moreover, it is easy to see that $\mathbf{C}(\boldsymbol{h}) = \mathbf{C}(\text{sample-}k(\boldsymbol{f} \circ \boldsymbol{\eta}))$; for instance,

$$\begin{aligned} c_{j,\mathrm{tp}}(\boldsymbol{h}) &= \mathbb{E}\left[\eta_j(\boldsymbol{x})\theta_j(\boldsymbol{x})\right] = \mathbb{E}\left[\mathbb{E}\left[\eta_j(\boldsymbol{x})\theta_j(\boldsymbol{x}) \,|\, \boldsymbol{\eta}(\boldsymbol{x}) = \boldsymbol{\eta}\right]\right] \\ &= \mathbb{E}\left[\eta_j \mathbb{E}\left[\theta_j(\boldsymbol{x}) \,|\, \boldsymbol{\eta}(\boldsymbol{x}) = \boldsymbol{\eta}\right]\right] = \mathbb{E}\left[\eta_j f_j(\boldsymbol{\eta})\right] \\ &= \mathbb{E}\left[\eta_j(\boldsymbol{x}) f_j(\boldsymbol{\eta}(\boldsymbol{x}))\right] = c_{j,\mathrm{tp}}(f_j \circ \boldsymbol{\eta}), \end{aligned} \tag{A.65}$$

etc. □

Hence, any confusion matrix achievable by some $\boldsymbol{h} = \text{sample-}k(\boldsymbol{\theta}(\boldsymbol{x}))$ is also achievable by some $\boldsymbol{h} = \text{sample-}k(\boldsymbol{f} \circ \boldsymbol{\eta})$, so that the set of achievable confusion matrices can be written as $\mathcal{C}_{\mathbb{P}}^{m,@k} = \{\mathbf{C}(\text{sample-}k(\boldsymbol{f} \circ \boldsymbol{\eta})) \colon \boldsymbol{f} \in \mathcal{F}\}$, where we denote $\mathcal{F} = \{\boldsymbol{f} \colon [0,1]^m \to \Delta^{@k}\}$. From this moment on, we thus, without loss of generality, we consider optimizing the metrics over functions $\boldsymbol{f} \in \mathcal{F}$ of random vector $\boldsymbol{\eta}$, and make the relation $\boldsymbol{\theta} = \boldsymbol{f} \circ \boldsymbol{\eta}$ implicit, writing $\Psi(\boldsymbol{f})$ for $\Psi(\text{sample-}k(\boldsymbol{f} \circ \boldsymbol{\eta}))$ and using $\mathbf{C}(\boldsymbol{f})$ to denote the confusion matrices $\mathbf{C}(\text{sample-}k(\boldsymbol{f} \circ \boldsymbol{\eta}))$, that is

$$\mathbf{C}(\boldsymbol{f}) = (\boldsymbol{c}_1(f_1), \dots, \boldsymbol{c}_m(f_m))\,, \tag{A.66}$$

where

$$\boldsymbol{c}_j(f_j) = \left[\underbrace{\mathbb{E}_{\boldsymbol{\eta}}[(1-\eta_j)(1-f_j(\boldsymbol{\eta}))]}_{\text{tn}}, \underbrace{\mathbb{E}_{\boldsymbol{\eta}}[\eta_j(1-f_j(\boldsymbol{\eta}))]}_{\text{fp}}, \underbrace{\mathbb{E}_{\boldsymbol{\eta}}[(1-\eta_j)f_j(\boldsymbol{\eta})]}_{\text{fn}}, \underbrace{\mathbb{E}_{\boldsymbol{\eta}}[\eta_j f_j(\boldsymbol{\eta})]}_{\text{tp}} \right]. \tag{A.67}$$

Lemma A.4.4. *Mapping $\boldsymbol{f} \mapsto \mathbf{C}(\boldsymbol{f})$ is continuous: for any $\boldsymbol{f}, \boldsymbol{f}' \in \mathcal{F}$*

$$\|\mathbf{C}(\boldsymbol{f}) - \mathbf{C}(\boldsymbol{f}')\|_F \leqslant \sqrt{2\mathbb{E}_{\boldsymbol{\eta}}\left[\|\boldsymbol{f}(\boldsymbol{\eta}) - \boldsymbol{f}'(\boldsymbol{\eta})\|_2^2\right]}, \tag{A.68}$$

where $\|\mathbf{C}(\boldsymbol{f}) - \mathbf{C}(\boldsymbol{f}')\|_F \coloneqq \sqrt{\sum_{j=1}^m \|\boldsymbol{c}_j(f_j) - \boldsymbol{c}_j(f_j')\|_F^2}$

Proof. Fix $j \in [m]$. Using $\delta_j(\boldsymbol{\eta}) = f_j(\boldsymbol{\eta}) - f_j'(\boldsymbol{\eta})$, we have from the definition:

$$\begin{aligned}
&\boldsymbol{c}_j(f_j) - \boldsymbol{c}_j(f_j') \\
&\quad = \Big[\mathbb{E}_{\boldsymbol{\eta}}[-(1-\eta_j)\delta_j(\boldsymbol{\eta})], \mathbb{E}_{\boldsymbol{\eta}}[-\eta_j\delta_j(\boldsymbol{\eta})], \mathbb{E}_{\boldsymbol{\eta}}[(1-\eta_j)\delta_j(\boldsymbol{\eta})], \mathbb{E}_{\boldsymbol{\eta}}[\eta_j\delta_j(\boldsymbol{\eta})]\Big] \\
&\quad = \mathbb{E}_{\boldsymbol{\eta}}\Big[\delta_j(\boldsymbol{\eta})\big[-(1-\eta_j), -\eta_j, 1-\eta_j, \eta_j\big]\Big].
\end{aligned} \tag{A.69}$$

Since the squared Frobenious norm $\boldsymbol{X} \mapsto \|\boldsymbol{X}\|_F^2$ is convex, we can use Jensen's inequality $\|\mathbb{E}[\boldsymbol{X}]\|_F^2 \leqslant \mathbb{E}\left[\|\boldsymbol{X}\|_F^2\right]$ to get

$$\begin{aligned}
\|\boldsymbol{c}_j(f_j) - \boldsymbol{c}_j(f_j')\|_F^2 &\leqslant \Big\|\mathbb{E}_{\boldsymbol{\eta}}\Big[\delta_j(\boldsymbol{\eta})\big[-(1-\eta_j), -\eta_j, 1-\eta_j, \eta_j\big]\Big]\Big\|_F^2 \\
&\leqslant \mathbb{E}_{\boldsymbol{\eta}}\left[\delta_j(\boldsymbol{\eta})^2 \left\|\big[-(1-\eta_j), -\eta_j, 1-\eta_j, \eta_j\big]\right\|_F^2\right] \\
&\leqslant 2\mathbb{E}_{\boldsymbol{\eta}}[\delta_j(\boldsymbol{\eta})^2],
\end{aligned} \tag{A.70}$$

where we used

$$\begin{aligned}
\left\|\big[-(1-\eta_j), -\eta_j, 1-\eta_j, \eta_j\big]\right\|_F^2 &= 2\left((1-\eta_j)^2 + \eta_j^2\right) \\
&\leqslant 2 \max_{x \in [0,1]} \left((1-x)^2 + x^2\right) = 2\,.
\end{aligned} \tag{A.71}$$

Summing the inequality over $j = 1, \dots, m$ and taking square root on both sides finishes the proof. □

We will now show that the set of achievable confusion matrices $\mathcal{C}_{\mathbb{P}}^{m,@k}$ is compact. To this end, we first prove a result from the functional analysis.

Lemma A.4.5. *Let $\mathcal{L} : \mathbb{H} \to \mathcal{V}$ be continuous affine operator between a Hilbert space $\mathbb{H}$ and a finite dimensional vector space $\mathcal{V}$. If $\mathcal{S} \subset \mathbb{H}$ is closed, bounded, and convex, then $\mathcal{L}(\mathcal{S})$ is compact.*

Proof. Observe that it suffices to prove this when $\mathcal{L}$ is linear since being compact is translation invariant.

The proof is inspired by an answer to a related question on Mathematics Stack Exchange. It suffices to prove every $\mathcal{L}(x_n)$ sequence in $\mathcal{L}$ has a convergent subsequence whose limit is in $\mathcal{L}(\mathcal{S})$.

By the Banach–Alaoglu theorem, balls in Hilbert spaces are weakly compact. For the convenience of the reader we will sketch this proof. Recall that weak convergence $x_n \to x$ in $\mathbb{H}$ means that for all linear functionals $\phi \in \mathbb{H}^\star$ there is convergence $\phi(x_n) \to \phi(x)$. Likewise weakly compact means every sequence has a subsequence that converges weakly. Now onto the proof.

Let x_n be a bounded sequence in $\mathbb{H}$. Let $\{e_1, e_2, \dots\}$ be a Hilbert basis for $\mathbb{H}$ and the dual vectors $\{\phi_1, \phi_2, \dots\}$ a Hilbert basis for $\mathbb{H}^\star$ where $\phi_i(x) = \langle x, e_i \rangle$. Now apply the diagonal proof method of the as in the Arzelà-Ascoli theorem of successively passing to subsequences. Since the sequence is bounded we know that $\phi_1(x_n)$ is bounded in $\mathbb{R}$ and hence we can extract a subsequence x_n so that $\phi_1(x_n) \to a_1$ where we may keep x_1. Now on this subsequence do the same for $\phi_2(x_n) \to a_2$ but keep x_1 and x_2. Continue this process, where at the m-th step one keeps the first m terms from the previous subsequence. The resulting diagonal subsequence x_n is such that $\phi_i(x_n) \to a_i$ for each $i = 1, 2, \dots$. The element $x = \sum_{i=1}^{\infty} a_i e_i$ is in $\mathbb{H}$ (by Bessel's inequality, the weak convergence results, and the fact that the original sequence was bounded). It remains to verify that $x_n \to x$ weakly, but for this it suffices to check $\phi_i(x_n) \to \phi_i(x)$ and by design this is the case.

Now, returning to the proof of the lemma since $\mathcal{S} \subset \mathbb{H}$ is bounded, it is contained in a ball, and hence by passing to a subsequence we have $x_n \to x$ in the weak topology for some $x \in \mathbb{H}$. Furthermore $x \in \mathcal{S}$. If if wasn't, then since $\mathcal{S}$ is closed and convex, by the Hahn–Banach separation theorem there is a separating hyperplane $\phi \in \mathbb{H}$ so $\phi(x) < \inf \mathcal{L}(\mathcal{S})$. But this contradicts that the weak convergence $x_n \to x$ since $x_n \in \mathcal{S}$.

So it remains to prove convergence $\mathcal{L}(x_n) \to \mathcal{L}(x)$. Since $\mathcal{L}$ is continuous we have convergence $\mathcal{L}(x_n) \to \mathcal{L}(x)$ in the weak topology, but this implies normal convergence since $\mathcal{V}$ is finite dimensional. □

Lemma A.4.6. *$\mathcal{C}_{\mathbb{P}}^{m,@k}$ is a compact set.*

Proof. To show that $\mathcal{C}_{\mathbb{P}}^{m,@k}$ is compact, we will invoke Lemma A.4.5. To place ourselves in its setting, let the Hilbert space be $\mathbb{H} = L^2([0,1]^m, \mathbb{R}^m, \mu)$ where μ is the probability measure on $[0,1]^m$ associated with random vector $\boldsymbol{\eta}(\boldsymbol{x})$. The

inner product for $\boldsymbol{f}, \boldsymbol{g} : [0,1]^m \to \mathbb{R}^m$ in $\mathbb{H}$ is

$$\langle \boldsymbol{f}, \boldsymbol{g} \rangle = \int_{[0,1]^m} \langle \boldsymbol{f}(\boldsymbol{\eta}), \boldsymbol{g}(\boldsymbol{\eta}) \rangle \, d\mu(\boldsymbol{\eta}) \tag{A.72}$$

where the inner product inside the integral is the normal dot product in $\mathbb{R}^m$.

We have the affine map defined via (7.3) and (7.5)

$$\mathcal{L} : \mathbb{H} \to \mathbb{R}^{m \times 4} \quad \text{where} \quad \mathcal{L}(\boldsymbol{f}) = \mathbf{C}(\boldsymbol{f} \circ \boldsymbol{\eta}) \,, \tag{A.73}$$

and let the subset $\mathcal{S} \subset \mathbb{H}$ be

$$\mathcal{S} = \{ \boldsymbol{f} \in \mathbb{H} : \boldsymbol{f}([0,1]^m) \subset \Delta^{@k} \text{ almost everywhere} \}. \tag{A.74}$$

Since $\mathcal{C}_{\mathbb{P}}^{m,@k} = \mathcal{L}(\mathcal{S})$, it suffices to verify the assumptions in Lemma A.4.5.

The map $\mathcal{L}$ is continuous by Lemma A.4.4. The set $\mathcal{S}$ is convex since the set of $\Delta^{@k} \subset \mathbb{R}^m$ is convex. Likewise for bounded using also that the we are working with a probability measure: If $\boldsymbol{f} \in \mathcal{S}$, then $\|\boldsymbol{f}(\boldsymbol{\eta})\|^2 \leqslant m$ for all $\boldsymbol{\eta} \in [0,1]^m$ and hence

$$\|\boldsymbol{f}\|^2 = \int_{[0,1]^m} \|\boldsymbol{f}(\boldsymbol{\eta})\|^2 d\mu(\boldsymbol{\eta}) \leqslant \int_{[0,1]^m} m \, d\mu(\boldsymbol{\eta}) = m. \tag{A.75}$$

Similarly, the closedness of $\Delta^{@k} \subset \mathbb{R}^m$ translates into the closedness of $\mathcal{S}$ as we now prove. Suppose there is a sequence of $\boldsymbol{f}_n \in \mathcal{S}$ with $\boldsymbol{f}_n \to \boldsymbol{f}$ and $\boldsymbol{f} \notin \mathcal{S}$. This means the set of points

$$A = \{ \boldsymbol{\eta} \in [0,1]^m : \boldsymbol{f}(\boldsymbol{\eta}) \notin \Delta^{@k} \} \tag{A.76}$$

that $\boldsymbol{f}$ maps out of $\Delta^{@k}$ has positive measure $\mu(A) > 0$. In $\mathbb{R}^m$ there is a well defined distance function $d(\boldsymbol{z}, \Delta^{@k}) = \inf_{\boldsymbol{v} \in \Delta^{@k}} \|\boldsymbol{z} - \boldsymbol{v}\|$ and for $\epsilon > 0$ define the set

$$A_\epsilon = \{ \boldsymbol{\eta} \in [0,1]^m : d(\boldsymbol{f}(\boldsymbol{\eta}), \Delta^{@k}) > \epsilon \} \tag{A.77}$$

Note that $A = \bigcup_{j=1}^{\infty} A_{1/j}$ since $\Delta^{@k}$ is closed and hence there is some $\epsilon > 0$ such that $\mu(A_\epsilon) > 0$. Therefore for all n

$$\begin{aligned} \|\boldsymbol{f} - \boldsymbol{f}_n\|^2 &= \int_{[0,1]^m} \|\boldsymbol{f}(\boldsymbol{\eta}) - \boldsymbol{f}_n(\boldsymbol{\eta})\|^2 d\mu(\boldsymbol{\eta}) \\ &\geqslant \int_{A_\epsilon} \|\boldsymbol{f}(\boldsymbol{\eta}) - \boldsymbol{f}_n(\boldsymbol{\eta})\|^2 d\mu(\boldsymbol{\eta}) \geqslant \int_{A_\epsilon} \epsilon^2 \, d\mu(\boldsymbol{\eta}) = \epsilon^2 \mu(A_\epsilon) > 0 \end{aligned} \tag{A.78}$$

where the second inequality uses that $f_n(\boldsymbol{\eta}) \in \Delta^{@k}$ almost everywhere. That $\|\boldsymbol{f} - \boldsymbol{f}_n\|^2$ is uniformly bounded away from 0 contradicts that $\boldsymbol{f}_n \to \boldsymbol{f}$ in $\mathbb{H}$. □

Theorem 7.2.4 (Regret for admissible multi-label utilities under the PU framework)**.** *Let the data distribution $\mathbb{P}[\boldsymbol{x}, \boldsymbol{y}]$ and metric Ψ satisfy Assumption 7.2.2 and Assumption 7.2.3, respectively. Then, there exists an optimal $\mathbf{C}^\star \in \mathcal{C}_{\mathbb{P}}^{m,@k}$, that is $\Psi(\mathbf{C}^\star) = \Psi^\star$. Moreover, any classifier $\boldsymbol{h}^\star$ maximizing the linear utility $\mathbf{G} \cdot \mathbf{C}(\boldsymbol{h})$ over $\boldsymbol{h} \in \mathcal{H}$ with $\mathbf{G} = [\boldsymbol{g}_1, \ldots, \mathbf{G}_m]$ given by $\boldsymbol{g}_j = \nabla_{\widehat{\boldsymbol{c}}_j} \Psi(\widehat{\boldsymbol{c}}^\star)$, also maximizes $\Psi(\boldsymbol{h})$ over $\boldsymbol{h} \in \mathcal{H}$.*

Proof. Let $\mathbf{C}^\star = \arg\max_{\mathbf{C}\in\mathcal{C}_{\mathbb{P}}^{m,@k}} \Psi(\mathbf{C})$, which exists by the compactness of $\mathcal{C}_{\mathbb{P}}^{m,@k}$ (Lemma A.4.6) and the continuity of Ψ. By the first order optimality and convexity of $\mathcal{C}_{\mathbb{P}}^{m,@k}$, for all $\mathbf{C} \in \mathcal{C}_{\mathbb{P}}^{m,@k}$

$$\nabla\Psi(\mathbf{C}^\star)\cdot\mathbf{C}^\star \geqslant \nabla\Psi(\mathbf{C}^\star)\cdot\mathbf{C}. \tag{A.79}$$

which implies:

$$\mathbf{C}^\star = \operatorname*{arg\,max}_{\mathbf{C}\in\mathcal{C}_{\mathbb{P}}^{m,@k}} \mathbf{G}\cdot\mathbf{C} \tag{A.80}$$

for $\mathbf{G} = \nabla\Psi(\mathbf{C}^\star)$.

We now show that $\mathbf{C}^\star$ is the unique optimizer of (A.80). Using Assumption 7.2.3 applied to $\mathbf{C}^\star$, for all $j \in [m]$:

$$\begin{aligned}\frac{\partial}{\partial\epsilon}\Psi(\boldsymbol{c}_1^\star,\ldots,\boldsymbol{c}_j^\star+\epsilon[1,-1,-1,1],\ldots,\boldsymbol{c}_m^\star)\Big|_{\epsilon=0} &= \nabla_{\mathbf{C}[j]}\Psi(\mathbf{C}^\star)\cdot[1,-1,-1,1]\\ &= g_{j,\mathrm{tn}} + g_{j,\mathrm{tp}} - g_{j,\mathrm{fp}} - g_{j,\mathrm{fn}} = a_j > 0\,,\end{aligned} \tag{A.81}$$

with coefficients $a_j, j \in [m]$ defined in Theorem Theorem 7.2.1.

Now, since we just showed that $a_j \neq 0$ for all j, and $\boldsymbol{\eta}$ has a density, coordinates of $\boldsymbol{a}\odot\boldsymbol{\eta}+\boldsymbol{b}$ are all distinct with probability one. This means that, with probability one, top-k $(\boldsymbol{a}\odot\boldsymbol{\eta}+\boldsymbol{b})$ is a singleton, and thus the optimizers of the linear utility $\mathbf{G}\cdot\mathbf{C}(\boldsymbol{h})$ can only differ on a zero measure set, so they all have the same confusion matrix. Thus, $\mathbf{C}^\star$ uniquely maximizes linear utility $\mathbf{G}\cdot\mathbf{C}$ over $\mathbf{C}\in\mathcal{C}_{\mathbb{P}}^{m,@k}$.

This means, however, that any classifier $\boldsymbol{h}^\star$ maximizing $\mathbf{G}\cdot\mathbf{C}(\boldsymbol{h})$ over $\boldsymbol{h}\in\mathcal{H}$ has $\mathbf{C}(\boldsymbol{h}^\star) = \mathbf{C}^\star$, and thus maximizes Ψ. □

A.4.4 Consistency of Frank-Wolfe

In this section, we provide the formal proof of consistency for the Frank-Wolfe algorithm. We prove convergence for a slightly modified version of Algorithm 7.1, in which we replace the line search in line 13 with a fixed schedule, setting

$$\alpha^i \leftarrow \frac{2}{i+1}\,. \tag{A.82}$$

But we found that using the linear search as in the Algorithm 7.1 results in much faster convergence of the algorithm.

VC-dimension lemma

Lemma 7.3.2 (VC dimension for linear top-k classifiers)**.** *For $\boldsymbol{\eta}:\mathcal{X}\longrightarrow[0,1]^m$, define the hypothesis class*

$$\mathcal{H}_{\boldsymbol{\eta}}^j := \bigcup_{\boldsymbol{a},\boldsymbol{b}\in\mathbb{R}^m} \{h:\mathcal{X}\longrightarrow\{0,1\} \,:\, h(\boldsymbol{x}) = \mathbb{1}\left[j\in\arg\text{top-}k\ (\boldsymbol{a}\odot\boldsymbol{\eta}+\boldsymbol{b})\right]\}\,. \tag{7.12}$$

The VC-complexity of this class is

$$VC(\mathcal{H}_{\boldsymbol{\eta}}^j) \leqslant 6m\log(em)\,. \tag{7.13}$$

Proof. For any given $\boldsymbol{a}, \boldsymbol{b}$, the hypothesis predicts one, $h^j(\boldsymbol{x}) = 1$, iff there exists a set of $m-k$ indices $\mathcal{I} \subset [m]$ with $|\mathcal{I}| = m-k$, $j \notin \mathcal{I}$, such that for all $i \in \mathcal{I}$ the score $a_i\eta_i + b_i \leqslant a_j\eta_j + b_j$ is not greater than the score of label j.

This computation can be realized as a two-layer network. In the first layer $\boldsymbol{z}$, we calculate an indicator to determine which labels' scores are below the threshold, that is $z_i = \mathbb{1}\left[(a_i - a_j)\eta_i + (b_i - b_j)\right]$. Then, for the output, we threshold the sum of all the intermediate units to determine if j is predicted:

$$h(\boldsymbol{x}) = o(\boldsymbol{z}) \coloneqq \mathbb{1}\left[\sum_{i \neq j} z_i \geqslant m-k\right]. \tag{A.83}$$

The resulting network has $2(m-1)$ edges and $m-1$ computation nodes. If we allow the output node to be more general – a generic linear threshold function, the VC-dimension of this extended function class $\mathcal{H}'$ can only grow. For this extended class, we can apply [Baum and Haussler, 1988, Corollary 3], which gives an upper bound for the VC-dimension of

$$\mathrm{VC}(\mathcal{H}^j) \leqslant \mathrm{VC}(\mathcal{H}') \leqslant 2(m-1+2(m-1))\log(e(m-1)) \leqslant 6m\log(em)\,. \tag{A.84}$$

□

Additional lemmas

Before going into the main proof of Theorem 7.3.1, we provide two more helper lemmas:

Lemma A.4.7 (Regret for Linear Macro Measures)**.** *Let* $\mathbf{G}$ *be a linear macro-measure, that is,*

$$\begin{aligned}\mathbf{G}(\boldsymbol{h};\boldsymbol{\eta}) = \frac{1}{m}\sum_{j=1}^{m} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\big[&g_{j,\mathrm{tn}}(1-\eta_j(\boldsymbol{x}))(1-h_j(\boldsymbol{x})) \\ &+ g_{j,\mathrm{fp}}(1-\eta_j(\boldsymbol{x}))h_j(\boldsymbol{x}) + g_{j,\mathrm{fn}}\eta_j(\boldsymbol{x})(1-h_j(\boldsymbol{x})) + g_{j,\mathrm{tp}}\eta_j(\boldsymbol{x})h_j(\boldsymbol{x})\big]\,.\end{aligned} \tag{A.85}$$

Let $\boldsymbol{h}^\star(\boldsymbol{x}) \coloneqq \arg\max_{\boldsymbol{h}} \mathbf{G}(\boldsymbol{h};\boldsymbol{\eta})$*, and* $\widehat{\boldsymbol{h}}(\boldsymbol{x}) \coloneqq \arg\max_{\boldsymbol{h}} \mathbf{G}(\boldsymbol{h};\widehat{\boldsymbol{\eta}})$*. Then*

$$\mathbf{G}(\boldsymbol{h}^\star;\boldsymbol{\eta}) - \mathbf{G}(\widehat{\boldsymbol{h}};\boldsymbol{\eta}) \leqslant \frac{1}{m}\max_j \|\boldsymbol{g}_j\|_1 \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\left[\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_1\right]. \tag{A.86}$$

Proof. As $\mathbf{G}(\boldsymbol{h};\boldsymbol{\eta})$ is an affine function in its second argument, we can simplify differences to

$$\begin{aligned}\mathbf{G}(\boldsymbol{h};\boldsymbol{\eta}) - \mathbf{G}(\boldsymbol{h};\widehat{\boldsymbol{\eta}}) &= \frac{1}{m}\sum_{j=1}^{m} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\big[-g_{j,\mathrm{tn}}(\eta_j - \widehat{\eta}_j)(1-h_j) - g_{j,\mathrm{fp}}(\eta_j - \widehat{\eta}_j)h_j \\ &\qquad + g_{j,\mathrm{fn}}(\eta_j - \widehat{\eta}_j)(1-h_j) + g_{j,\mathrm{tp}}(\eta_j - \widehat{\eta}_j)h_j\big] \\ &= \frac{1}{m}\sum_{j=1}^{m} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\left[(\eta_j - \widehat{\eta}_j)\left((g_{j,\mathrm{tp}} - g_{j,\mathrm{fp}})h_j + (g_{j,\mathrm{fn}} - g_{j,\mathrm{tn}})(1-h_j)\right)\right].\end{aligned} \tag{A.87}$$

We can use this property to bound the regret of $\widehat{\boldsymbol{h}}$ as

$$\begin{aligned}\mathbf{G}(\boldsymbol{h}^\star;\boldsymbol{\eta}) - \mathbf{G}(\widehat{\boldsymbol{h}};\boldsymbol{\eta}) &= \mathbf{G}(\boldsymbol{h}^\star;\boldsymbol{\eta}) - \mathbf{G}(\boldsymbol{h}^\star;\widehat{\boldsymbol{\eta}}) + \mathbf{G}(\boldsymbol{h}^\star;\widehat{\boldsymbol{\eta}}) - \mathbf{G}(\widehat{\boldsymbol{h}};\boldsymbol{\eta}) \\ &\leqslant \mathbf{G}(\boldsymbol{h}^\star;\boldsymbol{\eta}) - \mathbf{G}(\boldsymbol{h}^\star;\widehat{\boldsymbol{\eta}}) + \mathbf{G}(\widehat{\boldsymbol{h}};\widehat{\boldsymbol{\eta}}) - \mathbf{G}(\widehat{\boldsymbol{h}};\boldsymbol{\eta}) \\ &= \frac{1}{m}\sum_{j=1}^{m} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\left[(\eta_j - \widehat{\eta}_j)\left((g_{j,\mathrm{tp}} - g_{j,\mathrm{fp}})(h_j^\star - \widehat{h}_j) + (g_{j,\mathrm{fn}} - g_{j,\mathrm{tn}})(\widehat{h}_j - h_j^\star)\right)\right] \\ &= \frac{1}{m}\sum_{j=1}^{m} (g_{j,\mathrm{tp}} - g_{j,\mathrm{fp}} - g_{j,\mathrm{fn}} + g_{j,\mathrm{tn}})\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\left[(\eta_j - \widehat{\eta}_j)(h_j^\star - \widehat{h}_j)\right]. \end{aligned} \tag{A.88}$$

As $h_j \in [0,1]$, we can bound $(\eta_j - \widehat{\eta}_j)(h_j^\star - \widehat{h}_j) \leqslant |(\eta_j - \widehat{\eta}_j)|$, resulting in

$$\mathbf{G}(\boldsymbol{h}^\star;\boldsymbol{\eta}) - \mathbf{G}(\widehat{\boldsymbol{h}};\boldsymbol{\eta}) \leqslant \frac{1}{m}\sum_{j=1}^{m} (g_{j,\mathrm{tp}} - g_{j,\mathrm{fp}} - g_{j,\mathrm{fn}} + g_{j,\mathrm{tn}})\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\left[|\eta_j - \widehat{\eta}_j|\right]. \tag{A.89}$$

Using the notation of Theorem 7.2.1, we set $a_j = g_{j,\mathrm{tp}} - g_{j,\mathrm{fp}} - g_{j,\mathrm{fn}} + g_{j,\mathrm{tn}}$, so that we can further bound

$$\begin{aligned}\mathbf{G}(\boldsymbol{h}^\star;\boldsymbol{\eta}) - \mathbf{G}(\widehat{\boldsymbol{h}};\boldsymbol{\eta}) &\leqslant \frac{1}{m}\max_j |a_j| \sum_{j=1}^{m} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\left[|\eta_j - \widehat{\eta}_j|\right] \\ &= \frac{1}{m}\max_j |a_j| \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\left[\|\boldsymbol{\eta} - \widehat{\boldsymbol{\eta}}\|_1\right]. \end{aligned} \tag{A.90}$$

Using $\max_j |a_j| \leqslant \max_j \|\boldsymbol{g}_j\|_1$ yields the claim. □

Lemma A.4.8 (Uniform Convergence of multi-label Confusion Matrices)**.** *For $\boldsymbol{\eta} : \mathcal{X} \longrightarrow [0,1]^m$, let*

$$\mathcal{H}_{\boldsymbol{\eta}} \coloneqq \bigcup_{\boldsymbol{a},\boldsymbol{b}\in\mathbb{R}^m} \{\boldsymbol{h} : \mathcal{X} \longrightarrow \{0,1\}^m : \boldsymbol{h}(\boldsymbol{x}) = \arg\operatorname{top}\text{-}k\ \ \boldsymbol{a}\odot\boldsymbol{\eta} + \boldsymbol{b}\}, \tag{A.91}$$

and let $\mathcal{S} \in (\mathcal{X}\times\{0,1\}^m)^n$ be an i.i.d. sample. Then for any $\delta \in (0,1]$, with probability at least $1-\delta$, we have

$$\sup_{\boldsymbol{h}\in\mathcal{H}_{\boldsymbol{\eta}}} \|\mathbf{C}(\boldsymbol{h},\mathbb{P}) - \widehat{\mathbf{C}}(\boldsymbol{h},\mathcal{S})\|_\infty \leqslant \tilde{\mathcal{O}}\left(\sqrt{\frac{m\cdot\log m\cdot\log n - \log\delta}{n}}\right). \tag{A.92}$$

Proof. Instead of showing uniform convergence for the entries of the confusion matrix directly, we show it for accuracy (0-1-error) $\mathrm{acc}^j = c_{j,\mathrm{tp}} + c_{j,\mathrm{tn}}$, condition positive rate $\mathrm{pp}_j = c_{j,\mathrm{fp}} + c_{j,\mathrm{tp}}$ and predicted positive rate $\mathrm{cp}_j = c_{j,\mathrm{fn}} + c_{j,\mathrm{tp}}$ first for a fixed $j \in [m]$.

To handle accuracy and predicted positives, consider

$$\begin{aligned}
&\sup_{\boldsymbol{h}\in\mathcal{H}_{\boldsymbol{\eta}}} \left|\mathrm{acc}^j(\boldsymbol{h}, \mathbb{P}\left[\boldsymbol{x}, \boldsymbol{y}\right]) - \widehat{\mathrm{acc}}^j(\boldsymbol{h}, \mathcal{S})\right| \\
&\quad = \sup_{\boldsymbol{h}\in\mathcal{H}_{\boldsymbol{\eta}}} \left|\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left[y_{i,j} = h_j(\boldsymbol{x}_i)\right] - \mathbb{P}\left[y_j = h_j(\boldsymbol{x})\right]\right| \\
&\quad = \sup_{h\in\mathcal{H}^j_{\boldsymbol{\eta}}} \left|\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left[y_{i,j} = h(\boldsymbol{x}_i)\right] - \mathbb{P}\left[y_j = h(\boldsymbol{x})\right]\right| . \qquad \text{(A.93)}
\end{aligned}$$

From Lemma 7.3.2, we know the VC-dimension of $\mathcal{H}^j_{\boldsymbol{\eta}}$ is some finite number d, thus, we can employ a standard bound for the 0-1 error to get, with probability $1-\delta$, that

$$\sup_{\boldsymbol{h}\in\mathcal{H}_{\boldsymbol{\eta}}} \left|\mathrm{acc}^j(\boldsymbol{h}, \mathbb{P}\left[\boldsymbol{x}, \boldsymbol{y}\right]) - \widehat{\mathrm{acc}}^j(\boldsymbol{h}, \mathcal{S})\right| \leqslant \sqrt{\frac{2d\log(2en/d) + 2\log(4/\delta)}{n}} . \qquad \text{(A.94)}$$

As this bound holds for all distributions of targets $\boldsymbol{y}$, it holds in particular also for $\boldsymbol{y} \equiv 1$, in which case accuracy turns into predicted positive rate.

Finally, we can bound the error on the condition positive rate simply using Hoeffding's inequality, as it does not depend on the hypothesis h. We get, with probability $1-\delta$

$$\sup_{\boldsymbol{h}\in\mathcal{H}_{\boldsymbol{\eta}}} \left|\mathrm{pp}(\boldsymbol{h}, \mathbb{P}\left[\boldsymbol{x}, \boldsymbol{y}\right]) - \widehat{\mathrm{pp}}(\boldsymbol{h}, \mathcal{S})\right| \leqslant \sqrt{\frac{\log(\delta^{-1})}{2n}} . \qquad \text{(A.95)}$$

Now we can reconstruct the actual entries of the confusion matrix. For example, the true positive rate as $\mathrm{tp} = \frac{1-\mathrm{acc}-\mathrm{pp}-\mathrm{cp}}{2}$. Thus, we can union bound, with probability $1-\delta$

$$\sup_{\boldsymbol{h}\in\mathcal{H}_{\boldsymbol{\eta}}} \left|\mathrm{tp}_j(\boldsymbol{h}, \mathbb{P}\left[\boldsymbol{x}, \boldsymbol{y}\right]) - \widehat{\mathrm{tp}}_j(\boldsymbol{h}, \mathcal{S})\right| \leqslant \sqrt{\frac{2d\log(2en/d) + 2\log(8/\delta)}{n}} + \sqrt{\frac{\log(3/\delta)}{2n}} . \qquad \text{(A.96)}$$

Similar bounds can be constructed for the other entries. Taking a union bound over all m labels:

$$\begin{aligned}
&\sup_{\boldsymbol{h}\in\mathcal{H}_{\boldsymbol{\eta}}} \left\|\mathbf{C}(\boldsymbol{h}, \mathbb{P}\left[\boldsymbol{x}, \boldsymbol{y}\right]) - \widehat{\mathbf{C}}(\boldsymbol{h}, \mathcal{S})\right\|_\infty \\
&\quad \leqslant \sqrt{\frac{2d\log(2en/d) + 2\log(8m/\delta)}{n}} + \sqrt{\frac{\log(3m/\delta)}{2n}} \\
&\quad = \sqrt{\frac{12m\log(em)\log(en/(3m(\log(em))) + 2\log(8m/\delta)}{n}} + \sqrt{\frac{\log(3m/\delta)}{2n}} . \qquad \text{(A.97)}
\end{aligned}$$

In order to combine the two square-root terms, we can apply the arithmetic-

quadratic mean inequality, to arrive at the claimed bound

$$\sup_{\boldsymbol{h}\in\mathcal{H}_{\boldsymbol{\eta}}} \|\mathbf{C}(\boldsymbol{h},\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right]) - \widehat{\mathbf{C}}(\boldsymbol{h},\mathcal{S})\|_\infty \leqslant \sqrt{\frac{48m\log(em)\log(en/(3m(\log(em))) + 10\log(\sqrt[5]{3\cdot 8^4}m/\delta)}{2n}}\,. \tag{A.98}$$

Finally, using $3m(\log(em)) \geqslant 1$, we simplify this part

$$\log(en/(3m(\log(em))) \leqslant \log(en)\,, \tag{A.99}$$

which results in

$$\sup_{\boldsymbol{h}\in\mathcal{H}_{\boldsymbol{\eta}}} \|\mathbf{C}(\boldsymbol{h},\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right]) - \widehat{\mathbf{C}}(\boldsymbol{h},\mathcal{S})\|_\infty \leqslant \sqrt{\frac{\mathcal{O}(m\log m\log n) + \mathcal{O}(\log(m/\delta))}{n}}\,. \tag{A.100}$$

□

Bound for Linear Optimization Step

The preceding results allow to prove a bound on the approximation error for each linear optimization step that is performed as part of the Frank-Wolfe algorithm:

Lemma A.4.9. *Let $\Psi : \mathcal{C}^m \longrightarrow \mathbb{R}_{\geqslant 0}$ be concave over $\mathcal{C}^m_{\mathbb{P}}$, L-Lipschitz, and β-smooth w.r.t. the L_1-norm. Let $\boldsymbol{h} \in \mathcal{H}$ be some classifier, and denote $\mathbf{G} \coloneqq \nabla\Psi(\widehat{\mathbf{C}}(\boldsymbol{h},\mathcal{S}))$. Let $\widehat{\boldsymbol{g}}$ be the deterministic classifier that empirically optimizes the linear objective induced by Ψ according to Theorem 7.2.1. For two classifiers $\boldsymbol{h}'$ and $\boldsymbol{g}'$, define*

$$\mathfrak{L}_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h}',\boldsymbol{g}') \coloneqq \mathbf{C}(\boldsymbol{g}',\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right]) \cdot \nabla\Psi(\mathbf{C}(\boldsymbol{h}',\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right])), \tag{A.101}$$

Then for any $\delta \in (0,1]$, with probability at least $1-\delta$ (over draws of $\mathcal{S}$ from $\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right]^n$), we have

$$\mathfrak{L}_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h},\widehat{\boldsymbol{g}}) \geqslant \max_{\boldsymbol{g}'} \mathfrak{L}_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h},\boldsymbol{g}') - \epsilon_{\mathcal{S}}\,, \tag{A.102}$$

where

$$\epsilon_{\mathcal{S}} = 8L\frac{1}{m}\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_1 + 8m^2\beta \sup_{\boldsymbol{h}'\in\mathcal{H}} \tilde{\mathcal{O}}\left(\sqrt{\frac{m\cdot\log m\cdot\log n - \log\delta}{n}}\right). \tag{A.103}$$

Proof. Define an empirical counterpart to $\mathfrak{L}_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}$, the population-level utility of a classifier for an empirically estimated gradient, as

$$\mathfrak{L}_{\mathcal{S}}(\boldsymbol{h}',\boldsymbol{g}') \coloneqq \mathbf{C}(\boldsymbol{g}',\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right]) \cdot \nabla\Psi(\widehat{\mathbf{C}}(\boldsymbol{h}',\mathcal{S}))\,, \tag{A.104}$$

and the (population-level) optimal classifier $\boldsymbol{g}^\star \in \arg\max_{\boldsymbol{g}'} \mathfrak{L}_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h},\boldsymbol{g}')$ for the exact gradient, whose existence is guaranteed by Theorem 7.2.1. Then we can

write

$$\max_{\boldsymbol{g}'} \mathfrak{L}_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h},\boldsymbol{g}') - \mathfrak{L}_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h},\widehat{\boldsymbol{g}}) = \mathfrak{L}_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h},\boldsymbol{g}^\star) - \mathfrak{L}_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h},\widehat{\boldsymbol{g}})$$
$$= \mathfrak{L}_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h},\boldsymbol{g}^\star) - \mathfrak{L}_{\mathcal{S}}(\boldsymbol{h},\boldsymbol{g}^\star) + \mathfrak{L}_{\mathcal{S}}(\boldsymbol{h},\boldsymbol{g}^\star) - \mathfrak{L}_{\mathcal{S}}(\boldsymbol{h},\widehat{\boldsymbol{g}}) + \mathfrak{L}_{\mathcal{S}}(\boldsymbol{h},\widehat{\boldsymbol{g}}) - \mathfrak{L}_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h},\widehat{\boldsymbol{g}})\,. \tag{A.105}$$

Now we turn to bounding each of these terms. For the second, we get

$$\begin{aligned}\mathfrak{L}_{\mathcal{S}}(\boldsymbol{h},\boldsymbol{g}^\star) - \mathfrak{L}_{\mathcal{S}}(\boldsymbol{h},\widehat{\boldsymbol{g}}) &= \mathbf{C}(\boldsymbol{g}^\star,\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right])\cdot\mathbf{G} - \mathbf{C}(\widehat{\boldsymbol{g}},\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right])\cdot\mathbf{G}\\ &\leqslant \max_{\boldsymbol{g}'}\mathbf{C}(\boldsymbol{g}',\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right])\cdot\mathbf{G} - \mathbf{C}(\widehat{\boldsymbol{g}},\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right])\cdot\mathbf{G}\\ &\leqslant 2\frac{1}{m}\max_j \|\boldsymbol{g}_j\|_1 \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_1, \end{aligned}\tag{A.106}$$

where the last step used that $\widehat{\boldsymbol{g}}$ is the empirical maximizer of the linear measure corresponding to $\mathbf{G}$, in order to apply Lemma A.4.7. Now, if Ψ is L-Lipschitz w.r.t. the L_1-norm, then

$$\forall \mathbf{C}' :\ |\nabla\Psi(\mathbf{C})\cdot\mathbf{C}'| \leqslant \|\mathbf{C}'\|_1\,. \tag{A.107}$$

Let $j\in[m]$, and applying (A.107) to $\mathbf{C}' = \tilde{\mathbf{C}}$ for which $\tilde{\mathbf{C}}^i = \mathbf{0}$ for all $i\neq j$, and $\tilde{\mathbf{C}}^i = 0.25\cdot\mathbf{1}$, we get

$$0.25\boldsymbol{g}_j\cdot\mathbf{1} \leqslant L \ \Leftrightarrow\ \|\boldsymbol{g}_j\|_1 \leqslant 4L\,. \tag{A.108}$$

As this holds for all j, the upper bound turns into

$$\mathfrak{L}_{\mathcal{S}}(\boldsymbol{h},\boldsymbol{g}^\star) - \mathfrak{L}_{\mathcal{S}}(\boldsymbol{h},\widehat{\boldsymbol{g}}) \leqslant 8L\frac{1}{m}\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_1 \tag{A.109}$$

To bound the other two terms, we can use Hölder's inequality:

$$\begin{aligned}&\mathfrak{L}_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h},\boldsymbol{g}^\star) - \mathfrak{L}_{\mathcal{S}}(\boldsymbol{h},\boldsymbol{g}^\star)\\ &\quad= \mathbf{C}(\boldsymbol{g}^\star,\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right])\cdot\nabla\Psi(\mathbf{C}(\boldsymbol{h},\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right])) - \mathbf{C}(\boldsymbol{g}^\star,\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right])\cdot\nabla\Psi(\widehat{\mathbf{C}}(\boldsymbol{h},\mathcal{S}))\\ &\quad= \mathbf{C}(\boldsymbol{g}^\star,\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right])\cdot\Big(\nabla\Psi(\mathbf{C}(\boldsymbol{h},\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right])) - \nabla\Psi(\widehat{\mathbf{C}}(\boldsymbol{h},\mathcal{S}))\Big)\\ &\quad\leqslant \|\nabla\Psi(\mathbf{C}(\boldsymbol{h},\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right])) - \nabla\Psi(\widehat{\mathbf{C}}(\boldsymbol{h},\mathcal{S}))\|_\infty\cdot\|\mathbf{C}(\boldsymbol{g}^\star,\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right])\|_1 \qquad \text{(Hölder)}\\ &\quad= m\|\nabla\Psi(\mathbf{C}(\boldsymbol{h},\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right])) - \nabla\Psi(\widehat{\mathbf{C}}(\boldsymbol{h},\mathcal{S}))\|_\infty \qquad \text{(Normalization of } \mathbf{C})\\ &\quad\leqslant m\beta\|\mathbf{C}(\boldsymbol{h},\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right]) - \widehat{\mathbf{C}}(\boldsymbol{h},\mathcal{S})\|_1 \qquad (\beta\text{-smoothness})\\ &\quad\leqslant 4m^2\beta\|\mathbf{C}(\boldsymbol{h},\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right]) - \widehat{\mathbf{C}}(\boldsymbol{h},\mathcal{S})\|_\infty\\ &\quad\leqslant 4m^2\beta\sup_{\boldsymbol{h}'\in\mathcal{H}}\|\mathbf{C}(\boldsymbol{h}',\mathbb{P}\left[\boldsymbol{x},\boldsymbol{y}\right]) - \widehat{\mathbf{C}}(\boldsymbol{h}',\mathcal{S})]\|_\infty \end{aligned}\tag{A.110}$$

The same argument can be employed to bound the third term. Thus, applying

Lemma A.4.8, we get with probability at least $1-\delta$

$$\mathfrak{L}_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h},\boldsymbol{g}^\star) - \mathfrak{L}_{\mathbb{P}[\boldsymbol{x},\boldsymbol{y}]}(\boldsymbol{h},\widehat{\boldsymbol{g}}) \leqslant 8L\frac{1}{m}\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_1 + 8m^2\beta \sup_{\boldsymbol{h}'\in\mathcal{H}} \tilde{\mathcal{O}}\left(\sqrt{\frac{m\cdot\log m\cdot\log n - \log\delta}{n}}\right). \quad \text{(A.111)}$$

□

Consistency of fixed-step-schedule Frank-Wolfe

Theorem 7.3.1 (Consistency of the Frank-Wolfe algorithm)**.** *Assume the utility function $\Psi : [0,1]^{m\times 4} \longrightarrow \mathbb{R}$ is concave over $\mathcal{C}_{\mathbb{P}}^{m,@k}$, L-Lipschitz, and β-smooth w.r.t the L_1-norm. Let $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ be a sample drawn i.i.d. from $\mathbb{P}$. Further, let $\widehat{\boldsymbol{\eta}}$ be a label probability estimator learned from $\mathcal{S}_1$, and $\boldsymbol{h}_{\mathcal{S}}^{\mathrm{FW}}$ be the classifier obtained after κn iterations. Then, for any $\delta \in (0,1]$, with probability of at least $1-\delta$ overdraws of $\mathcal{S}$,*

$$\mathrm{Reg}(\boldsymbol{h}_{\mathcal{S}}^{\mathrm{FW}}) \leqslant \mathcal{O}\left(\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}[\boldsymbol{x}]}\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_1\right) + \tilde{\mathcal{O}}\left(m^2\sqrt{\frac{m\cdot\log m\cdot\log n - \log\delta}{n}}\right) + \frac{8\beta m}{\kappa n + 2}. \quad \text{(7.11)}$$

Proof. Define a curvature constant for the loss Ψ as

$$\begin{aligned} C_\Psi &\coloneqq \sup_{\mathbf{C}^1,\mathbf{C}^2\in\mathcal{C}_{\mathbb{P}}^m,\gamma\in[0,1]} \frac{2}{\gamma^2}\left(\Psi(\mathbf{C}^1+\gamma(\mathbf{C}^2-\mathbf{C}^1)) - \Psi(\mathbf{C}^1) - \gamma\left(\mathbf{C}^2-\mathbf{C}^1\right)\cdot\nabla\Psi(\mathbf{C}^1)\right) \\ &\leqslant \sup_{\mathbf{C}^1,\mathbf{C}^2\in\mathcal{C}_{\mathbb{P}}^m,\gamma\in[0,1]} \frac{2}{\gamma^2}\left(\frac{\beta}{2}\gamma^2\|\mathbf{C}^2-\mathbf{C}^1\|_1^2\right) \\ &= \beta \sup_{\mathbf{C}^1,\mathbf{C}^2\in\mathcal{C}_{\mathbb{P}}^m} \|\mathbf{C}^2-\mathbf{C}^1\|_1^2 \leqslant 4\beta m, \end{aligned} \quad \text{(A.112)}$$

and let $\epsilon_{\mathcal{S}}$ be defined as in Lemma A.4.9. Set $\delta_{\mathrm{apx}} = (t+1)\epsilon_{\mathcal{S}}/C_\Psi$ and $\hat{h}^i$ as in Algorithm 7.1. Let $\hat{f}^i$ be the classifier implicitly defined in iteration i, that is,

$$\hat{f}^i \coloneqq \sum_{j=1}^{i} \alpha^j \hat{h}^j. \quad \text{(A.113)}$$

For $1 \leqslant i \leqslant t$, we can apply Lemma A.4.9 to $\hat{f}^{i-1}$ and $\hat{h}^i$, which gives

$$\begin{aligned}
&\mathbf{C}(\hat{h}^i, \mathbb{P}[\boldsymbol{x}, \boldsymbol{y}]) \cdot \nabla\Psi(\mathbf{C}(\hat{f}^{i-1}, \mathbb{P}[\boldsymbol{x}, \boldsymbol{y}])) \\
&\quad \geqslant \max_{\boldsymbol{g}'} \mathbf{C}(\boldsymbol{g}', \mathbb{P}[\boldsymbol{x}, \boldsymbol{y}]) \cdot \nabla\Psi(\mathbf{C}(\hat{f}^{i-1}, \mathbb{P}[\boldsymbol{x}, \boldsymbol{y}])) - \epsilon_{\mathcal{S}} \\
&\quad = \max_{\mathbf{C} \in \mathcal{C}^m_{\mathbb{P}}} \mathbf{C} \cdot \nabla\Psi(\mathbf{C}(\hat{f}^{i-1}, \mathbb{P}[\boldsymbol{x}, \boldsymbol{y}])) - \epsilon_{\mathcal{S}} \\
&\quad = \max_{\mathbf{C} \in \overline{\mathcal{C}^m_{\mathbb{P}}}} \mathbf{C} \cdot \nabla\Psi(\mathbf{C}(\hat{f}^{i-1}, \mathbb{P}[\boldsymbol{x}, \boldsymbol{y}])) - \epsilon_{\mathcal{S}} \\
&\quad = \max_{\mathbf{C} \in \overline{\mathcal{C}^m_{\mathbb{P}}}} \mathbf{C} \cdot \nabla\Psi(\mathbf{C}(\hat{f}^{i-1}, \mathbb{P}[\boldsymbol{x}, \boldsymbol{y}])) - \frac{1}{2}\delta_{\text{apx}} \frac{2}{t+1} C_\Psi \\
&\quad \geqslant \max_{\mathbf{C} \in \overline{\mathcal{C}^m_{\mathbb{P}}}} \mathbf{C} \cdot \nabla\Psi(\mathbf{C}(\hat{f}^{i-1}, \mathbb{P}[\boldsymbol{x}, \boldsymbol{y}])) - \frac{1}{2}\delta_{\text{apx}} \frac{2}{i+1} C_\Psi \,. \qquad \text{(A.114)}
\end{aligned}$$

As we consider, for the proof, a Frank-Wolfe implementation with fixed step schedule $\frac{2}{i+1}$, the confusion matrices are related through

$$\mathbf{C}(\hat{f}^i, \mathbb{P}[\boldsymbol{x}, \boldsymbol{y}]) = \left(1 - \frac{2}{i+1}\right) \mathbf{C}(\hat{f}^{i-1}, \mathbb{P}[\boldsymbol{x}, \boldsymbol{y}]) + \frac{2}{i+1} \mathbf{C}(\hat{h}^i, \mathbb{P}[\boldsymbol{x}, \boldsymbol{y}]) \,. \quad \text{(A.115)}$$

With results (A.114) and (A.115), we now have the exact same situation as in Narasimhan et al. [2015, Proof of Theorem 16]. In particular, an application of Jaggi [2013, Theorem 1] gives the desired result. □

B

Technical details of the experiments and extended results

In this appendix, we provide more technical details of the experiments to ensure their replicability. We also provide results with other values of k as well as standard deviations of reported mean values.

B.1 Technical details of experimental setup

In the Table B.1, we present values of all hyperparameters used for training ensembles of probabilistic label trees (PLTs) using the napkinXC library [Jasinska-Kobus et al., 2020], as well as algorithms for the optimization of label-wise utilities, block coordinate ascent (BCA), and Frank-Wolfe, that were implemented in the xCOLUMNs library [Schultheis et al., 2023].

For training all PLTs, we used almost the same hyperparameters. Every ensemble consists of 3 trees, each tree is built by performing hierarchical balanced 2-means clustering on label representations obtained as the average of all their instances. The clustering is performed until the clusters reach a size smaller than 400. We represent instances using sparse TF-IDF features. Each tree node is learned using LIBLINEAR [Fan et al., 2008] that solves a dual L2-regularized logistic regression problem. We use two stopping criteria: a maximum number of iterations equal to 100 and a minimal improvement in the loss objective that we set to 0.01. After the training, the weights are sparsified by thresholding their absolute value at 0.01 to reduce the model size as in [Babbar and Schölkopf, 2017]. The cost parameter of the solver that corresponds to regularization strength is set to 32 for WikiLSHTC-325K, WikipediaLarge-500K, and Amazon-670K datasets and 16 for the rest. These values were recommended by [Jasinska-Kobus et al., 2020].

Similarly to PLTs, almost the same hyperparameters for BCA and Frank Wolfe algorithms, except constant υ that we add the denominator of the optimized metric. The best value of υ was selected using the validation set consisting of 30%

Table B.1: Hyperparameters of probabilistic label trees training and Block Coordinate Ascent and Frank Wolfe algorithms

Hyperparameter	Value(s)
PLTs training	
Number of trees in ensemble	3
Tree building method	hierarchical balanced 2-means clustering
Maximum number of leaves per node	400
LIBLINEAR solver	L2-regularized logistic regression (dual)
Regularization strength (cost)	$\{16, 32\}$
Minimal improvement in objective	0.01
Maximum number of iterations	100
Weights threshold	0.01
Block Coordinate Ascent	
Minimal improvement in objective (ϵ)	1e-7
Maximum number of iterations	100
Objective denominator constant υ	{1e-8, 1e-7, 1e-6, 1e-5}
Frank Wolfe	
Minimal improvement in objective (ϵ)	1e-6
Maximum number of iterations	100
Minimal contribution of the new classifier (ϵ_α)	1e-3
Objective denominator constant υ	{1e-8, 5e-7, 1e-7, 5e-6, 1e-6, 5e-5, 1e-5, 5e-4, 2e-4, 1e-4, 5e-3, 2e-3, 1e-3}

of the training set. The set of considered values of υ was bigger for the Frank Wolfe algorithm than for BCA.

The napkinXC library was implemented in C++ and Python, and xCOLUMNs was implemented in Python using numpy [Harris et al., 2020], Numba [Lam et al., 2015], autograd [Maclaurin et al., 2015] and PyTorch [Paszke et al., 2019] libraries.

The experiments were performed on machines with 64, 96, or 128 GB of RAM with Intel Xeon E5-2697, Intel Xeon Gold 5115, or Ryzen 3700X processors. All the experiments can be replicated on a machine with 64 GB of RAM.

The code for replicating the results of the experiments presented in this thesis can be found under `experiments` directories in the following two code repositories:

napkinXC https://github.com/mwydmuch/napkinXC

xCOLUMNs https://github.com/mwydmuch/xCOLUMNs

B.2 Extended results

B.2.1 Comparison of inference algorithms on datasets with synthetic labels

Table B.2: Results (%) for $k \in \{1, 3, 5\}$ on synthetic versions of XMLC datasets with ideal estimates of marginal conditional probabilities $\boldsymbol{\eta}(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}(\boldsymbol{x})$. Each experiments was repeated 5 times with mean and standard deviation reported after the $\pm$ sign. The green background indicates cells in which the inference algorithm matches the metric it optimizes and the gray text indicates results worse than those results for a given metric. The gray text indicates results worse than those with green background for a given metric. The best results are in **bold**, and the second best are in *italic*. * – because in this experiment we sample labels independently, Top-k becomes the optimal strategy for recall@k as showed in Theorem 3.3.1.

Method	Instance @1			Macro @1					
	P	PS	R	P	R	BA	F_1	JS	Cov
Synthetic RCV1x-2K									
Top-1	**90.62 ± 0.02**	*93.95 ± 0.45*	* **34.57 ± 0.01**	19.72 ± 0.13	1.49 ± 0.01	50.75 ± 0.00	2.11 ± 0.01	1.41 ± 0.01	24.80 ± 0.10
PS-1	*89.35 ± 0.02*	**96.21 ± 0.10**	*34.35 ± 0.01*	56.01 ± 0.06	19.38 ± 0.11	59.69 ± 0.05	22.39 ± 0.09	13.98 ± 0.07	93.48 ± 0.10
Pow-$1_{\beta=0.25}$	84.38 ± 0.01	93.05 ± 0.05	33.13 ± 0.01	49.16 ± 0.15	23.68 ± 0.08	61.84 ± 0.04	26.63 ± 0.07	16.64 ± 0.05	95.87 ± 0.15
Pow-$1_{\beta=0.5}$	76.73 ± 0.02	86.09 ± 0.05	30.87 ± 0.01	39.15 ± 0.11	29.20 ± 0.09	64.60 ± 0.05	26.82 ± 0.06	16.74 ± 0.04	97.00 ± 0.09
Log-1	85.37 ± 0.02	93.31 ± 0.04	33.29 ± 0.01	52.40 ± 0.13	15.82 ± 0.05	57.91 ± 0.03	21.54 ± 0.06	13.12 ± 0.04	90.75 ± 0.19
Macro-R_{prior}-1	51.89 ± 0.02	60.41 ± 0.05	20.72 ± 0.01	22.41 ± 0.07	32.36 ± 0.10	66.17 ± 0.05	17.24 ± 0.03	10.19 ± 0.02	96.85 ± 0.10
Macro-BA_{prior}-1	52.60 ± 0.02	61.17 ± 0.04	21.07 ± 0.01	22.50 ± 0.07	32.36 ± 0.10	66.17 ± 0.05	17.31 ± 0.03	10.23 ± 0.02	96.83 ± 0.09
BCA(Macro-P@1)	67.60 ± 0.02	69.33 ± 0.14	25.75 ± 0.01	**74.84 ± 0.79**	3.30 ± 0.10	51.64 ± 0.05	5.35 ± 0.15	3.07 ± 0.09	88.56 ± 0.95
BCA(Macro-R@1)	39.96 ± 0.02	47.79 ± 0.02	15.06 ± 0.01	20.24 ± 0.04	**33.07 ± 0.13**	**66.52 ± 0.06**	14.34 ± 0.03	8.36 ± 0.02	97.77 ± 0.21
BCA(Macro-BA@1)	41.09 ± 0.02	48.96 ± 0.02	15.64 ± 0.01	20.32 ± 0.03	*33.06 ± 0.13*	**66.52 ± 0.07**	14.40 ± 0.03	8.40 ± 0.02	97.77 ± 0.21
BCA(Macro-F_1@1)	73.91 ± 0.02	82.99 ± 0.05	30.03 ± 0.01	43.32 ± 0.10	26.74 ± 0.11	63.37 ± 0.05	**30.06 ± 0.10**	*18.65 ± 0.08*	*99.58 ± 0.10*
BCA(Macro-JS@1)	73.85 ± 0.02	82.97 ± 0.04	30.06 ± 0.01	41.49 ± 0.11	27.10 ± 0.09	63.55 ± 0.04	*29.87 ± 0.09*	**18.69 ± 0.07**	99.50 ± 0.13
BCA(Cov@1)	1.84 ± 0.01	3.87 ± 0.01	0.27 ± 0.00	17.78 ± 0.02	20.91 ± 0.07	60.43 ± 0.03	5.27 ± 0.02	2.83 ± 0.01	**99.85 ± 0.09**
FW(Macro-P@1)	63.96 ± 0.01	65.94 ± 0.08	24.49 ± 0.01	*68.78 ± 0.60*	3.50 ± 0.09	51.74 ± 0.05	5.46 ± 0.13	3.13 ± 0.08	82.26 ± 0.71
FW(Macro-R@1)	39.80 ± 0.01	47.60 ± 0.02	14.89 ± 0.01	20.18 ± 0.04	32.99 ± 0.13	*66.48 ± 0.07*	14.30 ± 0.02	8.36 ± 0.02	97.73 ± 0.15
FW(Macro-BA@1)	40.89 ± 0.02	48.74 ± 0.02	15.47 ± 0.01	20.27 ± 0.04	32.99 ± 0.13	*66.48 ± 0.07*	14.34 ± 0.02	8.39 ± 0.02	97.77 ± 0.15
FW(Macro-F_1@1)	73.90 ± 0.03	82.96 ± 0.04	30.02 ± 0.01	42.62 ± 0.11	26.58 ± 0.12	63.29 ± 0.06	29.66 ± 0.11	18.38 ± 0.09	99.25 ± 0.14
FW(Macro-JS@1)	73.98 ± 0.02	83.10 ± 0.04	30.12 ± 0.01	41.63 ± 0.08	26.95 ± 0.08	63.47 ± 0.04	29.71 ± 0.09	18.57 ± 0.07	99.17 ± 0.09

Method	Instance @1 P	Instance @1 PS	Instance @1 R	Macro @1 P	Macro @1 R	Macro @1 BA	Macro @1 F_1	Macro @1 JS	Macro @1 Cov
Synthetic EURLex-4K									
Top-1	**88.02 ± 0.10**	157.69 ± 0.18	* **21.73 ± 0.06**	28.12 ± 0.10	7.86 ± 0.03	53.93 ± 0.01	11.01 ± 0.04	7.29 ± 0.02	34.32 ± 0.07
PS-1	77.28 ± 0.10	**212.54 ± 0.32**	18.74 ± 0.02	55.25 ± 0.25	37.67 ± 0.12	68.83 ± 0.06	38.48 ± 0.14	28.58 ± 0.11	79.59 ± 0.16
Pow-1$_{\beta=0.25}$	78.42 ± 0.11	*211.44 ± 0.34*	19.03 ± 0.02	54.44 ± 0.28	35.66 ± 0.17	67.83 ± 0.08	37.17 ± 0.17	27.59 ± 0.13	76.94 ± 0.26
Pow-1$_{\beta=0.5}$	69.16 ± 0.15	202.41 ± 0.33	16.38 ± 0.04	52.40 ± 0.13	38.11 ± 0.13	69.05 ± 0.07	37.57 ± 0.10	27.69 ± 0.09	79.94 ± 0.11
Log-1	*81.71 ± 0.12*	204.82 ± 0.24	*19.95 ± 0.05*	50.47 ± 0.19	28.72 ± 0.22	64.36 ± 0.11	31.75 ± 0.14	23.33 ± 0.13	67.92 ± 0.24
Macro-R$_{\text{prior}}$-1	57.45 ± 0.19	186.15 ± 0.46	13.24 ± 0.06	46.92 ± 0.13	39.82 ± 0.17	69.90 ± 0.09	35.26 ± 0.14	25.51 ± 0.14	79.89 ± 0.08
Macro-BA$_{\text{prior}}$-1	57.49 ± 0.18	186.21 ± 0.45	13.26 ± 0.05	46.98 ± 0.12	39.82 ± 0.17	69.90 ± 0.09	35.27 ± 0.14	25.51 ± 0.14	79.96 ± 0.08
BCA(Macro-P@1)	39.03 ± 0.11	86.13 ± 0.43	8.85 ± 0.02	**67.60 ± 0.39**	18.30 ± 0.19	59.14 ± 0.09	21.82 ± 0.12	15.34 ± 0.09	78.53 ± 0.39
BCA(Macro-R@1)	57.73 ± 0.20	186.99 ± 0.52	13.36 ± 0.06	46.99 ± 0.13	**41.52 ± 0.13**	**70.76 ± 0.07**	36.19 ± 0.10	25.97 ± 0.10	82.49 ± 0.23
BCA(Macro-BA@1)	57.73 ± 0.18	186.95 ± 0.47	13.36 ± 0.06	46.88 ± 0.12	**41.52 ± 0.15**	*70.75 ± 0.07*	36.17 ± 0.11	25.96 ± 0.09	82.35 ± 0.22
BCA(Macro-F_1@1)	70.77 ± 0.13	203.94 ± 0.30	16.94 ± 0.03	57.01 ± 0.17	39.02 ± 0.09	69.50 ± 0.05	**41.38 ± 0.07**	*30.05 ± 0.08*	*86.34 ± 0.17*
BCA(Macro-JS@1)	70.45 ± 0.17	205.29 ± 0.39	16.83 ± 0.05	55.74 ± 0.26	39.60 ± 0.03	69.79 ± 0.01	*41.14 ± 0.12*	**30.22 ± 0.12**	85.48 ± 0.24
BCA(Cov@1)	26.37 ± 0.06	105.21 ± 0.19	5.98 ± 0.01	47.02 ± 0.06	33.26 ± 0.23	66.62 ± 0.11	24.94 ± 0.01	16.60 ± 0.03	**88.29 ± 0.22**
FW(Macro-P@1)	34.62 ± 0.09	80.83 ± 0.20	7.76 ± 0.01	*58.09 ± 0.16*	17.93 ± 0.22	58.96 ± 0.11	20.11 ± 0.14	14.23 ± 0.13	69.53 ± 0.27
FW(Macro-R@1)	57.55 ± 0.17	187.37 ± 0.47	13.32 ± 0.05	47.14 ± 0.06	*41.15 ± 0.16*	70.57 ± 0.08	35.92 ± 0.10	25.83 ± 0.08	82.12 ± 0.18
FW(Macro-BA@1)	57.44 ± 0.15	187.18 ± 0.43	13.28 ± 0.05	47.15 ± 0.07	41.14 ± 0.15	70.57 ± 0.08	35.92 ± 0.10	25.83 ± 0.08	82.15 ± 0.19
FW(Macro-F_1@1)	70.74 ± 0.14	204.75 ± 0.38	16.94 ± 0.03	56.36 ± 0.21	38.68 ± 0.12	69.33 ± 0.06	40.48 ± 0.13	29.47 ± 0.12	84.88 ± 0.23
FW(Macro-JS@1)	70.41 ± 0.14	205.93 ± 0.44	16.82 ± 0.04	55.40 ± 0.22	39.23 ± 0.15	69.61 ± 0.08	40.44 ± 0.14	29.73 ± 0.13	84.49 ± 0.10
Synthetic EURLex-4.3K									
Top-1	**93.71 ± 0.04**	134.19 ± 0.07	* **23.37 ± 0.05**	27.70 ± 0.15	6.27 ± 0.03	53.14 ± 0.01	9.20 ± 0.03	6.02 ± 0.02	32.15 ± 0.12
PS-1	85.12 ± 0.05	**175.98 ± 0.26**	20.95 ± 0.05	59.73 ± 0.26	43.82 ± 0.24	71.91 ± 0.12	43.92 ± 0.19	32.99 ± 0.18	84.24 ± 0.16
Pow-1$_{\beta=0.25}$	84.65 ± 0.04	*175.14 ± 0.18*	20.77 ± 0.05	59.13 ± 0.26	41.79 ± 0.26	70.90 ± 0.13	42.78 ± 0.22	32.14 ± 0.21	82.35 ± 0.18
Pow-1$_{\beta=0.5}$	79.05 ± 0.05	171.21 ± 0.17	19.15 ± 0.04	57.42 ± 0.31	45.31 ± 0.25	72.65 ± 0.12	43.99 ± 0.19	32.92 ± 0.18	85.02 ± 0.22
Log-1	*87.28 ± 0.05*	170.95 ± 0.19	*21.50 ± 0.05*	54.94 ± 0.12	32.72 ± 0.15	66.36 ± 0.08	35.97 ± 0.12	26.70 ± 0.13	72.31 ± 0.11
Macro-R$_{\text{prior}}$-1	70.47 ± 0.05	160.27 ± 0.19	16.83 ± 0.04	52.58 ± 0.17	46.82 ± 0.26	73.41 ± 0.13	41.42 ± 0.13	30.48 ± 0.13	85.55 ± 0.19
Macro-BA$_{\text{prior}}$-1	70.53 ± 0.05	160.34 ± 0.19	16.84 ± 0.04	52.60 ± 0.17	46.82 ± 0.26	73.41 ± 0.13	41.43 ± 0.13	30.48 ± 0.13	85.56 ± 0.19
BCA(Macro-P@1)	53.48 ± 0.05	80.46 ± 0.12	12.42 ± 0.01	**71.07 ± 0.07**	19.11 ± 0.21	59.55 ± 0.10	22.02 ± 0.16	15.54 ± 0.18	83.09 ± 0.17
BCA(Macro-R@1)	68.89 ± 0.03	157.54 ± 0.11	16.41 ± 0.02	51.52 ± 0.07	**47.79 ± 0.15**	**73.89 ± 0.07**	41.16 ± 0.03	30.06 ± 0.02	86.62 ± 0.13
BCA(Macro-BA@1)	68.95 ± 0.03	157.62 ± 0.11	16.43 ± 0.02	51.55 ± 0.07	**47.79 ± 0.15**	**73.89 ± 0.08**	41.17 ± 0.03	30.07 ± 0.02	86.62 ± 0.13
BCA(Macro-F_1@1)	79.81 ± 0.05	170.65 ± 0.09	19.39 ± 0.03	61.02 ± 0.12	45.11 ± 0.15	72.55 ± 0.07	**46.69 ± 0.10**	*34.44 ± 0.11*	*89.32 ± 0.15*
BCA(Macro-JS@1)	79.38 ± 0.08	171.26 ± 0.15	19.29 ± 0.04	59.77 ± 0.14	45.77 ± 0.20	72.88 ± 0.10	*46.35 ± 0.14*	**34.60 ± 0.12**	88.45 ± 0.19
BCA(Cov@1)	15.50 ± 0.04	56.11 ± 0.17	3.45 ± 0.01	42.43 ± 0.11	35.41 ± 0.20	67.69 ± 0.10	23.12 ± 0.07	15.36 ± 0.07	**90.87 ± 0.11**
FW(Macro-P@1)	55.60 ± 0.05	83.57 ± 0.10	12.97 ± 0.01	*64.16 ± 0.12*	18.71 ± 0.22	59.35 ± 0.11	20.83 ± 0.16	14.76 ± 0.16	76.04 ± 0.20
FW(Macro-R@1)	69.16 ± 0.03	158.37 ± 0.08	16.47 ± 0.03	51.71 ± 0.12	*47.47 ± 0.23*	*73.73 ± 0.12*	40.99 ± 0.09	29.96 ± 0.08	86.37 ± 0.17
FW(Macro-BA@1)	69.22 ± 0.03	158.44 ± 0.09	16.49 ± 0.03	51.73 ± 0.11	*47.47 ± 0.23*	*73.73 ± 0.12*	41.00 ± 0.09	29.97 ± 0.08	86.38 ± 0.18
FW(Macro-F_1@1)	79.72 ± 0.07	170.95 ± 0.11	19.37 ± 0.03	60.41 ± 0.11	44.70 ± 0.24	72.35 ± 0.12	45.82 ± 0.15	33.83 ± 0.13	88.07 ± 0.13
FW(Macro-JS@1)	79.36 ± 0.07	171.58 ± 0.16	19.27 ± 0.04	59.44 ± 0.24	45.30 ± 0.25	72.65 ± 0.12	45.73 ± 0.19	34.12 ± 0.17	87.51 ± 0.20

Method	Instance @1			Macro @1					
	P	PS	R	P	R	BA	F_1	JS	Cov
Synthetic AmazonCat-13K									
Top-1	**91.38** ± **0.01**	95.31 ± 0.07	* **25.88** ± **0.01**	22.66 ± 0.04	2.41 ± 0.01	51.20 ± 0.01	3.82 ± 0.01	2.25 ± 0.01	29.11 ± 0.07
PS-1	*86.03* ± *0.01*	**111.09** ± **0.05**	*24.49* ± *0.00*	55.02 ± 0.11	46.96 ± 0.12	73.48 ± 0.06	43.26 ± 0.11	31.13 ± 0.09	90.60 ± 0.13
Pow-$1_{\beta=0.25}$	75.45 ± 0.02	*104.44* ± *0.05*	21.37 ± 0.01	50.35 ± 0.11	48.78 ± 0.06	74.39 ± 0.03	44.25 ± 0.10	31.80 ± 0.08	89.81 ± 0.06
Pow-$1_{\beta=0.5}$	66.55 ± 0.01	96.67 ± 0.03	18.34 ± 0.01	44.51 ± 0.07	54.96 ± 0.11	77.48 ± 0.06	43.20 ± 0.08	30.74 ± 0.07	92.02 ± 0.11
Log-1	76.78 ± 0.01	102.90 ± 0.02	21.45 ± 0.01	51.44 ± 0.05	35.07 ± 0.02	67.54 ± 0.01	37.97 ± 0.03	26.97 ± 0.02	81.18 ± 0.05
Macro-R_{prior}-1	53.02 ± 0.01	82.40 ± 0.03	13.59 ± 0.00	35.70 ± 0.08	57.32 ± 0.08	78.66 ± 0.04	36.98 ± 0.06	25.60 ± 0.05	92.29 ± 0.10
Macro-BA_{prior}-1	53.24 ± 0.01	82.63 ± 0.03	13.70 ± 0.00	35.73 ± 0.08	57.32 ± 0.08	78.66 ± 0.04	36.99 ± 0.06	25.61 ± 0.05	92.29 ± 0.10
BCA(Macro-P@1)	77.19 ± 0.01	80.00 ± 0.51	21.25 ± 0.00	**72.72** ± **0.18**	10.94 ± 0.08	55.47 ± 0.04	13.17 ± 0.09	8.28 ± 0.06	87.38 ± 0.22
BCA(Macro-R@1)	49.13 ± 0.01	77.69 ± 0.03	12.36 ± 0.00	33.61 ± 0.07	**57.80** ± **0.09**	**78.90** ± **0.05**	34.35 ± 0.05	23.63 ± 0.03	92.90 ± 0.10
BCA(Macro-BA@1)	49.40 ± 0.01	77.97 ± 0.03	12.50 ± 0.00	33.64 ± 0.07	**57.80** ± **0.09**	**78.90** ± **0.05**	34.36 ± 0.05	23.63 ± 0.03	*92.91* ± *0.10*
BCA(Macro-F_1@1)	68.05 ± 0.01	96.66 ± 0.02	18.82 ± 0.00	51.94 ± 0.09	49.64 ± 0.08	74.82 ± 0.04	**47.26** ± **0.08**	33.80 ± 0.07	92.83 ± 0.07
BCA(Macro-JS@1)	67.14 ± 0.01	96.11 ± 0.03	18.44 ± 0.00	50.60 ± 0.08	50.28 ± 0.07	75.14 ± 0.03	*47.06* ± *0.07*	**33.97** ± **0.06**	92.82 ± 0.04
BCA(Cov@1)	4.00 ± 0.00	10.28 ± 0.02	0.77 ± 0.00	20.30 ± 0.04	30.42 ± 0.12	65.21 ± 0.06	11.02 ± 0.05	6.33 ± 0.03	**94.54** ± **0.17**
FW(Macro-P@1)	77.26 ± 0.01	80.15 ± 0.45	21.27 ± 0.00	*70.75* ± *0.17*	10.83 ± 0.07	55.42 ± 0.03	12.92 ± 0.08	8.14 ± 0.06	84.65 ± 0.20
FW(Macro-R@1)	49.28 ± 0.01	77.86 ± 0.03	12.42 ± 0.00	33.62 ± 0.07	*57.72* ± *0.09*	*78.86* ± *0.04*	34.31 ± 0.06	23.60 ± 0.04	92.87 ± 0.10
FW(Macro-BA@1)	49.51 ± 0.01	78.10 ± 0.03	12.53 ± 0.00	33.64 ± 0.07	*57.72* ± *0.09*	*78.86* ± *0.04*	34.32 ± 0.06	23.61 ± 0.04	92.88 ± 0.10
FW(Macro-F_1@1)	68.02 ± 0.01	96.63 ± 0.03	18.81 ± 0.00	51.75 ± 0.10	49.49 ± 0.10	74.75 ± 0.05	47.04 ± 0.09	33.64 ± 0.08	92.63 ± 0.09
FW(Macro-JS@1)	67.16 ± 0.01	96.12 ± 0.03	18.45 ± 0.00	50.54 ± 0.10	50.07 ± 0.12	75.03 ± 0.06	46.85 ± 0.10	*33.81* ± *0.09*	92.53 ± 0.08
Synthetic AmazonCat-14K									
Top-1	**87.66** ± **0.01**	92.56 ± 0.03	* **38.03** ± **0.01**	31.09 ± 0.09	3.88 ± 0.02	51.94 ± 0.01	6.14 ± 0.03	3.61 ± 0.02	41.17 ± 0.12
PS-1	*83.40* ± *0.01*	**103.46** ± **0.02**	*36.67* ± *0.00*	50.70 ± 0.06	41.55 ± 0.07	70.77 ± 0.03	39.67 ± 0.06	27.87 ± 0.05	87.46 ± 0.10
Pow-$1_{\beta=0.25}$	75.89 ± 0.02	99.11 ± 0.02	33.96 ± 0.00	45.07 ± 0.05	44.96 ± 0.09	72.48 ± 0.05	40.72 ± 0.06	28.57 ± 0.06	88.37 ± 0.10
Pow-$1_{\beta=0.5}$	66.51 ± 0.02	90.88 ± 0.03	29.45 ± 0.01	36.72 ± 0.06	51.35 ± 0.06	75.67 ± 0.03	37.73 ± 0.04	26.12 ± 0.05	90.43 ± 0.11
Log-1	78.84 ± 0.01	*99.56* ± *0.18*	34.89 ± 0.00	48.43 ± 0.07	33.02 ± 0.07	66.51 ± 0.04	36.08 ± 0.05	25.06 ± 0.05	81.06 ± 0.15
Macro-R_{prior}-1	43.19 ± 0.01	66.91 ± 0.02	12.96 ± 0.00	28.01 ± 0.05	54.25 ± 0.07	77.12 ± 0.03	30.29 ± 0.05	20.47 ± 0.05	91.12 ± 0.06
Macro-BA_{prior}-1	43.83 ± 0.01	67.56 ± 0.02	13.56 ± 0.00	28.03 ± 0.06	54.24 ± 0.07	77.12 ± 0.03	30.30 ± 0.05	20.48 ± 0.05	91.12 ± 0.06
BCA(Macro-P@1)	61.90 ± 0.01	63.75 ± 0.04	28.80 ± 0.01	**71.88** ± **0.30**	7.09 ± 0.13	53.54 ± 0.06	9.97 ± 0.13	6.35 ± 0.11	81.04 ± 0.31
BCA(Macro-R@1)	40.75 ± 0.01	63.80 ± 0.02	11.84 ± 0.00	26.79 ± 0.04	**54.90** ± **0.11**	**77.45** ± **0.05**	28.17 ± 0.05	18.98 ± 0.04	91.76 ± 0.09
BCA(Macro-BA@1)	40.85 ± 0.01	63.91 ± 0.02	11.87 ± 0.00	26.83 ± 0.04	**54.90** ± **0.11**	**77.45** ± **0.06**	28.19 ± 0.05	18.99 ± 0.04	*91.79* ± *0.09*
BCA(Macro-F_1@1)	69.30 ± 0.01	92.06 ± 0.03	30.27 ± 0.01	47.67 ± 0.06	43.92 ± 0.10	71.96 ± 0.05	**43.45** ± **0.07**	*30.64* ± *0.08*	90.14 ± 0.05
BCA(Macro-JS@1)	69.41 ± 0.01	92.44 ± 0.02	30.73 ± 0.00	46.55 ± 0.08	44.49 ± 0.11	72.25 ± 0.06	*43.32* ± *0.09*	**30.77** ± **0.09**	90.13 ± 0.08
BCA(Cov@1)	2.66 ± 0.01	7.14 ± 0.02	0.67 ± 0.00	19.16 ± 0.02	28.12 ± 0.09	64.06 ± 0.05	7.71 ± 0.02	4.32 ± 0.02	**93.07** ± **0.05**
FW(Macro-P@1)	60.34 ± 0.01	62.30 ± 0.04	28.20 ± 0.01	*64.40* ± *0.20*	6.90 ± 0.07	53.45 ± 0.04	9.14 ± 0.07	5.78 ± 0.06	73.03 ± 0.20
FW(Macro-R@1)	40.56 ± 0.01	63.61 ± 0.02	11.69 ± 0.00	26.75 ± 0.05	*54.77* ± *0.07*	*77.38* ± *0.04*	28.08 ± 0.05	18.91 ± 0.04	91.66 ± 0.06
FW(Macro-BA@1)	40.81 ± 0.01	63.88 ± 0.02	11.89 ± 0.00	26.79 ± 0.05	*54.77* ± *0.07*	*77.38* ± *0.04*	28.10 ± 0.05	18.92 ± 0.04	91.65 ± 0.05
FW(Macro-F_1@1)	69.19 ± 0.01	91.94 ± 0.02	30.20 ± 0.01	47.33 ± 0.07	43.68 ± 0.13	71.84 ± 0.07	43.03 ± 0.09	30.32 ± 0.09	89.55 ± 0.11
FW(Macro-JS@1)	69.43 ± 0.01	92.45 ± 0.03	30.74 ± 0.00	46.37 ± 0.06	44.27 ± 0.12	72.13 ± 0.06	42.98 ± 0.08	30.51 ± 0.08	89.61 ± 0.05

Method	Instance @1			Macro @1					
	P	PS	R	P	R	BA	F_1	JS	Cov
Synthetic Wiki10-31K									
Top-1	**94.41** ± **0.11**	108.45 ± 0.22	* **9.36** ± **0.02**	1.84 ± 0.01	0.15 ± 0.00	50.07 ± 0.00	0.25 ± 0.00	0.14 ± 0.00	1.99 ± 0.00
PS-1	69.96 ± 0.17	**337.30** ± **1.13**	7.19 ± 0.01	25.85 ± 0.09	21.38 ± 0.08	60.69 ± 0.04	20.22 ± 0.09	15.94 ± 0.08	35.98 ± 0.08
Pow-1$_{\beta=0.25}$	67.91 ± 0.13	*334.49* ± *0.94*	6.70 ± 0.01	25.88 ± 0.07	20.79 ± 0.08	60.40 ± 0.04	19.90 ± 0.08	15.63 ± 0.08	35.99 ± 0.07
Pow-1$_{\beta=0.5}$	60.46 ± 0.14	330.52 ± 1.12	5.59 ± 0.01	25.68 ± 0.10	21.71 ± 0.10	60.85 ± 0.05	20.35 ± 0.10	15.99 ± 0.09	37.18 ± 0.07
Log-1	68.82 ± 0.18	318.48 ± 1.01	6.47 ± 0.02	24.78 ± 0.10	17.98 ± 0.06	58.99 ± 0.03	17.89 ± 0.06	13.86 ± 0.05	33.81 ± 0.10
Macro-R$_{\text{prior}}$-1	50.50 ± 0.19	316.33 ± 1.47	4.23 ± 0.01	23.48 ± 0.15	21.86 ± 0.11	60.93 ± 0.06	19.69 ± 0.12	15.35 ± 0.11	36.30 ± 0.11
Macro-BA$_{\text{prior}}$-1	50.56 ± 0.19	316.41 ± 1.45	4.25 ± 0.01	23.49 ± 0.15	21.86 ± 0.11	60.93 ± 0.06	19.69 ± 0.12	15.35 ± 0.11	36.30 ± 0.11
BCA(Macro-P@1)	71.14 ± 0.12	258.30 ± 0.63	7.29 ± 0.02	**58.82** ± **0.16**	17.87 ± 0.06	58.94 ± 0.03	23.85 ± 0.08	16.97 ± 0.08	*62.40* ± *0.14*
BCA(Macro-R@1)	48.93 ± 0.08	288.38 ± 0.59	4.04 ± 0.01	25.88 ± 0.05	**26.66** ± **0.03**	**63.33** ± **0.01**	23.40 ± 0.06	18.73 ± 0.06	40.15 ± 0.04
BCA(Macro-BA@1)	48.98 ± 0.07	288.38 ± 0.47	4.05 ± 0.01	25.93 ± 0.05	*26.65* ± *0.03*	*63.32* ± *0.02*	23.42 ± 0.05	18.74 ± 0.05	40.17 ± 0.05
BCA(Macro-F_1@1)	53.01 ± 0.17	287.63 ± 1.00	4.52 ± 0.02	48.62 ± 0.23	24.01 ± 0.07	62.01 ± 0.04	**28.59** ± **0.11**	*21.30* ± *0.08*	60.12 ± 0.19
BCA(Macro-JS@1)	52.36 ± 0.16	292.80 ± 0.78	4.43 ± 0.02	41.08 ± 0.16	25.57 ± 0.03	62.79 ± 0.02	*27.85* ± *0.08*	**21.40** ± **0.06**	53.43 ± 0.12
BCA(Cov@1)	51.30 ± 0.14	262.10 ± 0.95	4.22 ± 0.01	*53.95* ± *0.20*	19.94 ± 0.09	59.97 ± 0.04	25.88 ± 0.12	18.40 ± 0.10	**66.82** ± **0.09**
FW(Macro-P@1)	*74.98* ± *0.19*	254.96 ± 1.27	*7.68* ± *0.02*	38.36 ± 0.21	16.69 ± 0.15	58.34 ± 0.07	19.42 ± 0.16	14.34 ± 0.16	44.83 ± 0.19
FW(Macro-R@1)	48.83 ± 0.15	299.19 ± 1.11	4.03 ± 0.01	25.58 ± 0.09	24.66 ± 0.05	62.33 ± 0.03	22.05 ± 0.07	17.46 ± 0.07	39.22 ± 0.07
FW(Macro-BA@1)	48.86 ± 0.16	299.21 ± 1.11	4.04 ± 0.01	25.59 ± 0.09	24.66 ± 0.06	62.33 ± 0.03	22.05 ± 0.08	17.45 ± 0.07	39.23 ± 0.06
FW(Macro-F_1@1)	52.65 ± 0.12	298.27 ± 1.27	4.46 ± 0.01	37.68 ± 0.13	22.36 ± 0.10	61.18 ± 0.05	24.35 ± 0.10	18.32 ± 0.08	50.18 ± 0.11
FW(Macro-JS@1)	52.42 ± 0.08	304.27 ± 0.82	4.44 ± 0.01	34.44 ± 0.12	23.66 ± 0.09	61.83 ± 0.05	24.38 ± 0.09	18.84 ± 0.08	46.72 ± 0.13
Synthetic WikiLSHTC-325K									
Top-1	**63.44** ± **0.01**	127.36 ± 0.03	* **35.44** ± **0.00**	22.58 ± 0.02	17.64 ± 0.02	58.82 ± 0.01	16.98 ± 0.02	11.72 ± 0.02	42.45 ± 0.03
PS-1	56.78 ± 0.01	**155.91** ± **0.02**	32.24 ± 0.01	36.67 ± 0.02	43.42 ± 0.03	71.71 ± 0.02	35.31 ± 0.03	25.45 ± 0.02	73.57 ± 0.06
Pow-1$_{\beta=0.25}$	55.59 ± 0.01	*154.55* ± *0.04*	31.52 ± 0.01	35.66 ± 0.04	40.96 ± 0.04	70.48 ± 0.02	34.05 ± 0.04	24.52 ± 0.03	71.01 ± 0.06
Pow-1$_{\beta=0.5}$	51.94 ± 0.01	152.85 ± 0.05	29.57 ± 0.01	35.54 ± 0.03	45.60 ± 0.04	72.80 ± 0.02	35.72 ± 0.03	25.63 ± 0.02	75.58 ± 0.07
Log-1	*58.81* ± *0.01*	146.18 ± 0.05	*33.09* ± *0.00*	30.50 ± 0.03	28.46 ± 0.02	64.23 ± 0.01	25.98 ± 0.02	18.34 ± 0.02	57.76 ± 0.04
Macro-R$_{\text{prior}}$-1	47.57 ± 0.02	147.14 ± 0.04	27.20 ± 0.01	33.44 ± 0.03	47.12 ± 0.04	73.56 ± 0.02	34.85 ± 0.03	24.77 ± 0.03	76.95 ± 0.06
Macro-BA$_{\text{prior}}$-1	47.57 ± 0.02	147.14 ± 0.04	27.20 ± 0.01	33.44 ± 0.03	47.12 ± 0.04	73.56 ± 0.02	34.85 ± 0.03	24.77 ± 0.03	76.95 ± 0.06
BCA(Macro-P@1)	30.74 ± 0.01	65.41 ± 0.07	14.37 ± 0.00	**50.17** ± **0.06**	24.68 ± 0.06	62.34 ± 0.03	24.57 ± 0.05	17.47 ± 0.04	71.22 ± 0.08
BCA(Macro-R@1)	43.48 ± 0.01	138.63 ± 0.05	24.94 ± 0.01	31.52 ± 0.03	**48.48** ± **0.04**	**74.24** ± **0.02**	33.00 ± 0.03	22.98 ± 0.03	78.97 ± 0.06
BCA(Macro-BA@1)	43.49 ± 0.01	138.64 ± 0.05	24.94 ± 0.01	31.52 ± 0.03	**48.48** ± **0.04**	**74.24** ± **0.02**	33.00 ± 0.03	22.98 ± 0.03	78.97 ± 0.06
BCA(Macro-F_1@1)	51.27 ± 0.02	147.58 ± 0.03	29.29 ± 0.00	40.61 ± 0.04	44.71 ± 0.03	72.36 ± 0.02	**38.98** ± **0.04**	*28.10* ± *0.03*	*79.57* ± *0.07*
BCA(Macro-JS@1)	51.24 ± 0.01	148.27 ± 0.03	29.28 ± 0.01	40.13 ± 0.04	45.01 ± 0.04	72.50 ± 0.02	*38.87* ± *0.04*	**28.11** ± **0.03**	79.27 ± 0.08
BCA(Cov@1)	16.69 ± 0.00	71.56 ± 0.06	8.01 ± 0.00	30.90 ± 0.02	40.84 ± 0.05	70.42 ± 0.03	24.68 ± 0.03	16.40 ± 0.02	**81.69** ± **0.08**
FW(Macro-P@1)	30.08 ± 0.01	64.78 ± 0.07	13.87 ± 0.00	*46.76* ± *0.05*	24.40 ± 0.06	62.20 ± 0.03	23.43 ± 0.05	16.63 ± 0.04	67.24 ± 0.07
FW(Macro-R@1)	43.47 ± 0.02	138.84 ± 0.06	24.93 ± 0.01	31.56 ± 0.03	*48.18* ± *0.05*	*74.09* ± *0.02*	32.82 ± 0.04	22.92 ± 0.03	78.63 ± 0.06
FW(Macro-BA@1)	43.47 ± 0.02	138.84 ± 0.06	24.93 ± 0.01	31.56 ± 0.03	*48.18* ± *0.05*	*74.09* ± *0.02*	32.82 ± 0.04	22.92 ± 0.03	78.63 ± 0.06
FW(Macro-F_1@1)	51.23 ± 0.01	147.78 ± 0.02	29.27 ± 0.01	40.05 ± 0.03	44.24 ± 0.04	72.12 ± 0.02	38.15 ± 0.03	27.46 ± 0.03	78.41 ± 0.08
FW(Macro-JS@1)	51.21 ± 0.01	148.43 ± 0.03	29.26 ± 0.01	39.68 ± 0.02	44.57 ± 0.03	72.29 ± 0.02	38.14 ± 0.02	27.55 ± 0.01	78.25 ± 0.06

Method	Instance @1			Macro @1					
	P	PS	R	P	R	BA	F_1	JS	Cov
Synthetic WikipediaLarge-500K									
Top-1	**69.48** ± **0.02**	147.17 ± 0.05	* **30.30** ± **0.01**	23.73 ± 0.02	13.28 ± 0.01	56.64 ± 0.00	14.57 ± 0.01	10.01 ± 0.01	38.33 ± 0.02
PS-1	60.57 ± 0.02	**197.01** ± **0.07**	27.03 ± 0.01	46.17 ± 0.03	41.28 ± 0.01	70.64 ± 0.00	37.96 ± 0.02	27.67 ± 0.02	79.33 ± 0.03
Pow-$1_{\beta=0.25}$	60.34 ± 0.02	*195.54* ± *0.09*	26.75 ± 0.01	44.82 ± 0.03	38.41 ± 0.01	69.21 ± 0.01	36.25 ± 0.02	26.40 ± 0.02	75.89 ± 0.03
Pow-$1_{\beta=0.5}$	56.08 ± 0.01	194.14 ± 0.06	24.72 ± 0.01	45.16 ± 0.04	43.16 ± 0.01	71.58 ± 0.00	38.54 ± 0.02	27.95 ± 0.02	81.76 ± 0.04
Log-1	*64.40* ± *0.02*	181.89 ± 0.06	*28.30* ± *0.01*	37.31 ± 0.03	25.38 ± 0.01	62.69 ± 0.01	26.48 ± 0.02	18.81 ± 0.02	59.74 ± 0.03
Macro-R_{prior}-1	51.04 ± 0.01	187.33 ± 0.06	22.16 ± 0.01	42.68 ± 0.03	44.60 ± 0.01	72.30 ± 0.00	37.85 ± 0.01	27.15 ± 0.01	83.44 ± 0.03
Macro-BA_{prior}-1	51.05 ± 0.01	187.33 ± 0.06	22.16 ± 0.01	42.68 ± 0.03	44.60 ± **0.01**	72.30 ± 0.00	37.85 ± 0.01	27.15 ± 0.01	83.44 ± 0.03
BCA(Macro-P@1)	38.85 ± 0.02	93.96 ± 0.05	14.00 ± 0.01	**62.71** ± **0.04**	23.43 ± 0.02	61.71 ± 0.01	26.82 ± 0.02	18.53 ± 0.02	83.24 ± 0.03
BCA(Macro-R@1)	46.59 ± 0.01	175.97 ± 0.05	19.94 ± 0.01	41.02 ± 0.03	**46.89** ± **0.01**	**73.45** ± 0.00	36.88 ± 0.02	25.90 ± 0.02	86.69 ± 0.01
BCA(Macro-BA@1)	46.60 ± 0.01	175.98 ± 0.05	19.95 ± 0.01	41.02 ± 0.03	**46.89** ± **0.01**	**73.45** ± 0.00	36.88 ± 0.02	25.90 ± 0.02	86.69 ± 0.01
BCA(Macro-F_1@1)	54.48 ± 0.02	185.85 ± 0.06	24.08 ± 0.01	51.53 ± 0.03	43.67 ± 0.01	71.84 ± 0.00	**42.69** ± **0.02**	*30.69* ± *0.02*	*90.38* ± *0.02*
BCA(Macro-JS@1)	54.44 ± 0.02	187.35 ± 0.08	24.09 ± 0.01	50.42 ± 0.04	44.31 ± 0.01	72.15 ± 0.01	*42.47* ± *0.03*	**30.79** ± **0.03**	89.28 ± 0.03
BCA(Cov@1)	27.31 ± 0.02	117.19 ± 0.06	10.07 ± 0.01	41.01 ± 0.03	40.04 ± 0.02	70.02 ± 0.01	29.84 ± 0.02	19.81 ± 0.02	**92.42** ± **0.02**
FW(Macro-P@1)	38.21 ± 0.02	93.61 ± 0.06	13.58 ± 0.01	*57.94* ± *0.03*	23.01 ± 0.01	61.51 ± 0.00	25.43 ± 0.02	17.63 ± 0.02	77.29 ± 0.02
FW(Macro-R@1)	46.59 ± 0.01	176.67 ± 0.05	19.93 ± 0.01	41.05 ± 0.03	*46.28* ± *0.01*	*73.14* ± *0.01*	36.54 ± 0.01	25.69 ± 0.01	86.07 ± 0.01
FW(Macro-BA@1)	46.59 ± 0.01	176.67 ± 0.05	19.93 ± 0.01	41.05 ± 0.03	*46.28* ± *0.01*	*73.14* ± *0.00*	36.54 ± 0.01	25.69 ± 0.01	86.07 ± 0.01
FW(Macro-F_1@1)	54.45 ± 0.02	186.53 ± 0.08	24.06 ± 0.01	50.96 ± 0.04	42.93 ± 0.01	71.47 ± 0.00	41.53 ± 0.02	29.87 ± 0.02	88.56 ± 0.03
FW(Macro-JS@1)	54.43 ± 0.02	187.94 ± 0.08	24.08 ± 0.01	50.08 ± 0.03	43.59 ± 0.01	71.79 ± 0.00	41.52 ± 0.02	30.09 ± 0.02	87.81 ± 0.03
Synthetic Amazon-670K									
Top-1	**53.48** ± **0.03**	285.61 ± 0.17	* **18.13** ± **0.01**	19.30 ± 0.01	14.63 ± 0.01	57.32 ± 0.01	14.74 ± 0.01	10.52 ± 0.01	31.92 ± 0.02
PS-1	48.56 ± 0.03	**360.89** ± **0.25**	16.48 ± 0.02	24.82 ± 0.01	23.52 ± 0.01	61.76 ± 0.00	21.05 ± 0.01	15.57 ± 0.01	42.08 ± 0.01
Pow-$1_{\beta=0.25}$	50.02 ± 0.04	358.33 ± 0.28	16.99 ± 0.02	25.23 ± 0.01	22.47 ± 0.01	61.23 ± 0.00	20.72 ± 0.01	15.27 ± 0.01	41.99 ± 0.01
Pow-$1_{\beta=0.5}$	47.69 ± 0.02	*359.75* ± *0.16*	16.16 ± 0.01	24.44 ± 0.01	23.82 ± 0.01	61.91 ± 0.00	20.98 ± 0.00	15.52 ± 0.01	41.88 ± 0.02
Log-1	*52.47* ± *0.02*	332.94 ± 0.15	*17.79* ± *0.01*	23.66 ± 0.00	18.84 ± 0.01	59.42 ± 0.00	18.41 ± 0.01	13.35 ± 0.01	38.73 ± 0.02
Macro-R_{prior}-1	45.57 ± 0.03	355.03 ± 0.19	15.33 ± 0.02	23.08 ± 0.01	24.24 ± 0.01	62.12 ± 0.00	20.54 ± 0.01	15.18 ± 0.01	40.73 ± 0.02
Macro-BA_{prior}-1	45.57 ± 0.03	355.03 ± 0.19	15.33 ± 0.02	23.08 ± 0.01	24.24 ± **0.01**	62.12 ± 0.00	20.54 ± 0.01	15.18 ± 0.01	40.73 ± 0.02
BCA(Macro-P@1)	39.19 ± 0.03	261.11 ± 0.27	12.26 ± 0.01	**53.50** ± **0.07**	18.87 ± 0.03	59.43 ± 0.01	23.85 ± 0.03	16.59 ± 0.02	61.96 ± 0.08
BCA(Macro-R@1)	43.88 ± 0.03	332.55 ± 0.30	14.70 ± 0.01	24.03 ± 0.03	**26.54** ± **0.02**	**63.27** ± **0.01**	22.11 ± 0.02	16.35 ± 0.02	43.32 ± 0.03
BCA(Macro-BA@1)	43.88 ± 0.03	332.55 ± 0.30	14.70 ± 0.01	24.03 ± 0.03	**26.54** ± **0.02**	**63.27** ± **0.01**	22.11 ± 0.02	16.35 ± 0.02	43.32 ± 0.03
BCA(Macro-F_1@1)	46.94 ± 0.04	330.54 ± 0.31	15.84 ± 0.01	46.57 ± 0.05	23.94 ± 0.02	61.97 ± 0.01	**28.04** ± **0.03**	*19.82* ± *0.02*	*65.55* ± *0.06*
BCA(Macro-JS@1)	46.69 ± 0.02	336.72 ± 0.19	15.75 ± 0.01	41.71 ± 0.03	24.97 ± 0.01	62.48 ± 0.01	*27.61* ± *0.02*	**19.89** ± **0.01**	60.34 ± 0.03
BCA(Cov@1)	41.40 ± 0.04	303.57 ± 0.27	14.06 ± 0.02	*47.37* ± *0.04*	22.35 ± 0.02	61.17 ± 0.01	26.49 ± 0.02	18.26 ± 0.02	**69.00** ± **0.04**
FW(Macro-P@1)	38.06 ± 0.02	256.13 ± 0.24	11.74 ± 0.01	38.63 ± 0.04	17.70 ± 0.02	58.85 ± 0.01	20.29 ± 0.02	14.27 ± 0.02	49.60 ± 0.03
FW(Macro-R@1)	43.84 ± 0.03	339.51 ± 0.32	14.69 ± 0.01	23.86 ± 0.03	*25.47* ± *0.02*	*62.74* ± *0.01*	21.33 ± 0.02	15.77 ± 0.02	42.20 ± 0.03
FW(Macro-BA@1)	43.84 ± 0.03	339.52 ± 0.33	14.69 ± 0.01	23.86 ± 0.03	*25.47* ± *0.02*	*62.74* ± *0.01*	21.34 ± 0.02	15.77 ± 0.02	42.20 ± 0.03
FW(Macro-F_1@1)	46.80 ± 0.03	336.53 ± 0.27	15.78 ± 0.01	38.77 ± 0.03	23.06 ± 0.02	61.53 ± 0.01	24.91 ± 0.02	17.70 ± 0.02	57.08 ± 0.02
FW(Macro-JS@1)	46.62 ± 0.05	343.55 ± 0.42	15.74 ± 0.02	36.15 ± 0.05	24.08 ± 0.04	62.04 ± 0.02	25.04 ± 0.04	18.12 ± 0.03	53.92 ± 0.05

Method	Instance @3			Macro @3					
	P	PS	R	P	R	BA	F_1	JS	Cov
Synthetic RCV1x-2K									
Top-3	**73.88** ± **0.01**	*76.57* ± *0.12*	* **68.05** ± **0.02**	31.58 ± 0.10	8.39 ± 0.04	54.18 ± 0.02	9.85 ± 0.04	6.39 ± 0.03	56.74 ± 0.07
PS-3	*73.24* ± *0.01*	**77.68** ± **0.11**	*67.82* ± *0.02*	48.64 ± 0.08	36.52 ± 0.11	68.24 ± 0.06	34.84 ± 0.07	22.59 ± 0.06	99.38 ± 0.08
Pow-3$_{\beta=0.25}$	70.47 ± 0.01	76.06 ± 0.04	66.43 ± 0.02	42.75 ± 0.10	43.98 ± 0.12	71.97 ± 0.06	38.68 ± 0.08	25.23 ± 0.07	99.55 ± 0.07
Pow-3$_{\beta=0.5}$	65.39 ± 0.01	71.55 ± 0.03	63.16 ± 0.01	33.15 ± 0.09	52.36 ± 0.17	76.16 ± 0.09	35.09 ± 0.09	22.47 ± 0.07	99.76 ± 0.07
Log-3	70.98 ± 0.01	76.26 ± 0.13	66.59 ± 0.02	46.79 ± 0.07	34.11 ± 0.09	67.04 ± 0.04	35.62 ± 0.07	22.99 ± 0.06	97.90 ± 0.10
Macro-R$_{\text{prior}}$-3	45.99 ± 0.01	52.06 ± 0.05	46.07 ± 0.01	20.16 ± 0.03	57.95 ± 0.20	78.94 ± 0.10	21.01 ± 0.03	12.74 ± 0.02	99.84 ± 0.09
Macro-BA$_{\text{prior}}$-3	46.74 ± 0.01	52.80 ± 0.05	46.86 ± 0.01	20.24 ± 0.03	57.94 ± 0.20	78.94 ± 0.10	21.11 ± 0.03	12.80 ± 0.02	99.84 ± 0.09
BCA(Macro-P@3)	47.38 ± 0.01	47.85 ± 0.11	45.47 ± 0.00	**74.97** ± **0.64**	4.88 ± 0.13	52.41 ± 0.06	7.15 ± 0.15	4.23 ± 0.10	92.39 ± 0.58
BCA(Macro-R@3)	37.70 ± 0.01	43.45 ± 0.03	37.59 ± 0.01	19.14 ± 0.03	**58.74** ± **0.15**	**79.33** ± **0.08**	18.56 ± 0.02	11.19 ± 0.01	*99.85* ± *0.07*
BCA(Macro-BA@3)	38.69 ± 0.01	44.44 ± 0.02	38.69 ± 0.01	19.20 ± 0.03	**58.74** ± **0.16**	**79.33** ± **0.08**	18.63 ± 0.02	11.24 ± 0.01	*99.85* ± *0.07*
BCA(Macro-F_1@3)	65.89 ± 0.01	71.66 ± 0.06	63.27 ± 0.01	46.14 ± 0.07	44.47 ± 0.11	72.21 ± 0.05	**43.16** ± **0.05**	*28.52* ± *0.04*	99.72 ± 0.10
BCA(Macro-JS@3)	66.21 ± 0.01	72.00 ± 0.11	63.50 ± 0.01	45.57 ± 0.07	44.61 ± 0.09	72.29 ± 0.05	*43.04* ± *0.06*	**28.54** ± **0.04**	99.68 ± 0.09
BCA(Cov@3)	1.28 ± 0.00	2.61 ± 0.00	0.53 ± 0.00	12.54 ± 0.04	36.95 ± 0.12	68.41 ± 0.06	4.88 ± 0.01	2.60 ± 0.01	**99.90** ± **0.06**
FW(Macro-P@3)	43.58 ± 0.01	43.99 ± 0.06	41.74 ± 0.00	*72.28* ± *0.52*	5.74 ± 0.11	52.84 ± 0.06	7.05 ± 0.13	4.15 ± 0.08	89.44 ± 0.50
FW(Macro-R@3)	37.66 ± 0.01	43.41 ± 0.03	37.50 ± 0.01	19.13 ± 0.03	*58.68* ± *0.13*	*79.30* ± *0.06*	18.54 ± 0.02	11.18 ± 0.01	*99.85* ± *0.07*
FW(Macro-BA@3)	38.66 ± 0.01	44.42 ± 0.03	38.64 ± 0.01	19.19 ± 0.03	*58.68* ± *0.14*	*79.30* ± *0.07*	18.61 ± 0.02	11.23 ± 0.01	*99.85* ± *0.07*
FW(Macro-F_1@3)	65.88 ± 0.01	71.64 ± 0.05	63.27 ± 0.01	46.07 ± 0.05	44.30 ± 0.11	72.13 ± 0.06	43.01 ± 0.05	28.42 ± 0.04	99.58 ± 0.14
FW(Macro-JS@3)	66.21 ± 0.01	71.98 ± 0.10	63.50 ± 0.01	45.59 ± 0.05	44.47 ± 0.13	72.21 ± 0.07	42.95 ± 0.06	28.48 ± 0.05	99.58 ± 0.10
Synthetic EURLex-4K									
Top-3	**77.15** ± **0.04**	147.00 ± 0.11	* **53.96** ± **0.05**	45.07 ± 0.11	31.58 ± 0.07	65.78 ± 0.03	34.83 ± 0.05	25.77 ± 0.08	63.89 ± 0.20
PS-3	73.32 ± 0.05	**164.76** ± **0.13**	51.24 ± 0.06	55.31 ± 0.09	59.76 ± 0.16	79.87 ± 0.08	54.55 ± 0.06	41.88 ± 0.06	87.79 ± 0.08
Pow-3$_{\beta=0.25}$	73.22 ± 0.06	*164.37* ± *0.18*	51.17 ± 0.04	55.66 ± 0.12	58.73 ± 0.14	79.35 ± 0.07	54.38 ± 0.07	41.84 ± 0.08	87.21 ± 0.13
Pow-3$_{\beta=0.5}$	68.32 ± 0.04	160.79 ± 0.15	47.52 ± 0.03	52.08 ± 0.11	61.82 ± 0.08	80.90 ± 0.04	53.23 ± 0.06	40.50 ± 0.06	88.22 ± 0.10
Log-3	*74.30* ± *0.06*	163.25 ± 0.13	*51.93* ± *0.06*	55.90 ± 0.21	55.20 ± 0.11	77.59 ± 0.06	52.89 ± 0.11	40.70 ± 0.10	85.00 ± 0.15
Macro-R$_{\text{prior}}$-3	55.22 ± 0.08	142.87 ± 0.23	37.85 ± 0.05	43.01 ± 0.13	63.87 ± 0.13	81.92 ± 0.07	45.13 ± 0.08	32.80 ± 0.09	89.43 ± 0.09
Macro-BA$_{\text{prior}}$-3	55.30 ± 0.08	142.98 ± 0.24	37.91 ± 0.05	43.07 ± 0.14	63.87 ± 0.13	81.92 ± 0.07	45.17 ± 0.08	32.83 ± 0.09	89.45 ± 0.09
BCA(Macro-P@3)	26.79 ± 0.05	50.17 ± 0.13	17.74 ± 0.04	**67.92** ± **0.10**	23.94 ± 0.28	61.95 ± 0.14	26.48 ± 0.15	18.95 ± 0.13	83.59 ± 0.20
BCA(Macro-R@3)	54.60 ± 0.08	141.86 ± 0.22	37.36 ± 0.04	41.75 ± 0.03	*64.77* ± *0.13*	*82.37* ± *0.07*	44.21 ± 0.03	31.87 ± 0.02	*89.99* ± *0.15*
BCA(Macro-BA@3)	54.66 ± 0.08	141.96 ± 0.21	37.41 ± 0.04	41.75 ± 0.03	**64.78** ± **0.14**	**82.38** ± **0.07**	44.25 ± 0.03	31.89 ± 0.02	89.97 ± 0.15
BCA(Macro-F_1@3)	70.67 ± 0.05	161.30 ± 0.14	49.22 ± 0.04	56.94 ± 0.13	59.70 ± 0.18	79.84 ± 0.09	**55.98** ± **0.10**	*43.13* ± *0.09*	88.45 ± 0.19
BCA(Macro-JS@3)	70.64 ± 0.05	161.59 ± 0.14	49.17 ± 0.01	56.81 ± 0.13	59.77 ± 0.15	79.87 ± 0.07	*55.94* ± *0.08*	**43.19** ± **0.08**	88.38 ± 0.17
BCA(Cov@3)	12.85 ± 0.04	50.06 ± 0.19	8.65 ± 0.02	35.65 ± 0.14	45.42 ± 0.16	72.68 ± 0.08	21.19 ± 0.08	13.47 ± 0.06	**90.96** ± **0.19**
FW(Macro-P@3)	26.48 ± 0.04	50.12 ± 0.13	17.49 ± 0.03	*65.16* ± *0.10*	24.01 ± 0.23	61.98 ± 0.12	25.67 ± 0.16	18.40 ± 0.14	80.96 ± 0.10
FW(Macro-R@3)	54.46 ± 0.06	141.78 ± 0.17	37.28 ± 0.03	41.70 ± 0.07	64.49 ± 0.10	82.23 ± 0.05	43.90 ± 0.04	31.62 ± 0.04	89.82 ± 0.14
FW(Macro-BA@3)	53.97 ± 0.05	141.10 ± 0.15	36.89 ± 0.03	41.44 ± 0.06	64.49 ± 0.11	82.23 ± 0.06	43.68 ± 0.05	31.43 ± 0.04	89.78 ± 0.14
FW(Macro-F_1@3)	70.67 ± 0.04	161.28 ± 0.15	49.24 ± 0.01	56.93 ± 0.13	59.40 ± 0.15	79.69 ± 0.07	55.70 ± 0.09	42.88 ± 0.09	88.28 ± 0.13
FW(Macro-JS@3)	70.69 ± 0.06	161.71 ± 0.18	49.23 ± 0.01	56.83 ± 0.13	59.48 ± 0.16	79.73 ± 0.08	55.71 ± 0.09	42.99 ± 0.08	88.11 ± 0.11

Method	Instance @3			Macro @3					
	P	PS	R	P	R	BA	F_1	JS	Cov
Synthetic EURLex-4.3K									
Top-3	**86.11** ± 0.04	128.59 ± 0.12	* **60.79** ± 0.07	45.22 ± 0.14	27.94 ± 0.13	63.97 ± 0.06	32.20 ± 0.14	23.84 ± 0.15	60.75 ± 0.12
PS-3	*82.76* ± *0.09*	**143.28** ± 0.18	*58.49* ± *0.07*	59.93 ± 0.13	67.41 ± 0.16	83.70 ± 0.08	60.58 ± 0.09	47.80 ± 0.08	90.43 ± 0.14
Pow-3$_{\beta=0.25}$	81.82 ± 0.09	*142.85* ± *0.15*	57.82 ± 0.07	59.90 ± 0.14	66.90 ± 0.14	83.44 ± 0.07	60.48 ± 0.10	47.81 ± 0.11	89.94 ± 0.14
Pow-3$_{\beta=0.5}$	78.33 ± 0.08	140.31 ± 0.14	55.28 ± 0.06	56.68 ± 0.16	69.89 ± 0.14	84.94 ± 0.07	59.38 ± 0.10	46.51 ± 0.12	90.92 ± 0.09
Log-3	82.67 ± 0.07	142.08 ± 0.13	58.40 ± 0.07	60.10 ± 0.18	61.79 ± 0.19	80.89 ± 0.09	58.28 ± 0.14	46.02 ± 0.12	87.23 ± 0.15
Macro-R$_{\text{prior}}$-3	68.02 ± 0.05	127.88 ± 0.11	47.37 ± 0.05	47.24 ± 0.12	72.15 ± 0.20	86.06 ± 0.10	51.11 ± 0.10	38.48 ± 0.11	91.88 ± 0.17
Macro-BA$_{\text{prior}}$-3	68.18 ± 0.05	128.07 ± 0.11	47.51 ± 0.04	47.32 ± 0.13	72.15 ± 0.20	86.06 ± 0.10	51.19 ± 0.10	38.54 ± 0.11	91.88 ± 0.17
BCA(Macro-P@3)	38.85 ± 0.03	52.98 ± 0.06	26.06 ± 0.02	**71.27** ± 0.13	23.89 ± 0.19	61.93 ± 0.10	25.92 ± 0.08	18.63 ± 0.08	86.46 ± 0.19
BCA(Macro-R@3)	66.05 ± 0.07	125.14 ± 0.20	45.88 ± 0.05	45.61 ± 0.17	**72.66** ± 0.14	**86.32** ± 0.07	49.38 ± 0.13	36.86 ± 0.13	*92.25* ± *0.17*
BCA(Macro-BA@3)	66.23 ± 0.07	125.34 ± 0.19	46.04 ± 0.05	45.66 ± 0.17	**72.66** ± 0.14	**86.32** ± 0.07	49.43 ± 0.13	36.91 ± 0.13	*92.25* ± *0.17*
BCA(Macro-F_1@3)	80.01 ± 0.08	140.88 ± 0.18	56.42 ± 0.06	61.31 ± 0.09	67.80 ± 0.13	83.89 ± 0.06	**62.04** ± 0.04	*49.17* ± *0.05*	91.02 ± 0.18
BCA(Macro-JS@3)	79.80 ± 0.10	140.82 ± 0.19	56.28 ± 0.07	61.15 ± 0.09	67.89 ± 0.14	83.94 ± 0.07	*61.97* ± *0.06*	**49.22** ± 0.07	90.95 ± 0.19
BCA(Cov@3)	8.02 ± 0.03	28.04 ± 0.09	5.23 ± 0.01	29.06 ± 0.06	48.86 ± 0.23	74.40 ± 0.12	19.11 ± 0.08	12.13 ± 0.07	**92.64** ± 0.17
FW(Macro-P@3)	38.73 ± 0.03	52.79 ± 0.08	25.75 ± 0.01	*69.01* ± *0.23*	23.68 ± 0.14	61.82 ± 0.07	25.28 ± 0.07	18.18 ± 0.07	83.96 ± 0.13
FW(Macro-R@3)	66.20 ± 0.06	125.34 ± 0.15	45.98 ± 0.04	45.33 ± 0.14	*72.43* ± *0.20*	*86.20* ± *0.10*	49.03 ± 0.12	36.56 ± 0.12	92.13 ± 0.17
FW(Macro-BA@3)	66.36 ± 0.06	125.54 ± 0.14	46.11 ± 0.04	45.37 ± 0.14	*72.43* ± *0.19*	*86.20* ± *0.10*	49.08 ± 0.12	36.60 ± 0.12	92.13 ± 0.17
FW(Macro-F_1@3)	79.98 ± 0.09	140.88 ± 0.18	56.40 ± 0.06	61.26 ± 0.08	67.51 ± 0.18	83.75 ± 0.09	61.83 ± 0.06	48.99 ± 0.04	90.77 ± 0.16
FW(Macro-JS@3)	79.79 ± 0.09	140.84 ± 0.17	56.27 ± 0.06	61.10 ± 0.08	67.64 ± 0.18	83.81 ± 0.09	61.79 ± 0.06	49.06 ± 0.04	90.73 ± 0.18
Synthetic AmazonCat-13K									
Top-3	**77.60** ± 0.01	83.98 ± 0.04	* **60.10** ± 0.01	42.03 ± 0.16	17.83 ± 0.05	58.91 ± 0.02	21.53 ± 0.06	14.47 ± 0.04	63.11 ± 0.14
PS-3	*75.74* ± *0.01*	**89.77** ± 0.04	*58.93* ± *0.01*	51.27 ± 0.09	64.23 ± 0.12	82.11 ± 0.06	53.30 ± 0.10	39.34 ± 0.09	93.31 ± 0.08
Pow-3$_{\beta=0.25}$	70.10 ± 0.01	*86.98* ± *0.05*	55.34 ± 0.01	47.50 ± 0.09	68.91 ± 0.12	84.45 ± 0.06	53.82 ± 0.09	39.66 ± 0.08	93.14 ± 0.10
Pow-3$_{\beta=0.5}$	63.03 ± 0.01	80.67 ± 0.04	49.80 ± 0.01	39.67 ± 0.09	74.77 ± 0.14	87.38 ± 0.07	48.71 ± 0.11	34.84 ± 0.09	94.21 ± 0.14
Log-3	70.60 ± 0.01	86.74 ± 0.02	55.48 ± 0.01	50.21 ± 0.12	58.80 ± 0.08	79.40 ± 0.04	52.12 ± 0.10	38.68 ± 0.08	89.00 ± 0.08
Macro-R$_{\text{prior}}$-3	46.15 ± 0.01	63.28 ± 0.03	35.08 ± 0.00	27.48 ± 0.05	78.28 ± 0.15	89.13 ± 0.07	35.38 ± 0.06	24.05 ± 0.05	94.65 ± 0.14
Macro-BA$_{\text{prior}}$-3	46.52 ± 0.01	63.64 ± 0.03	35.51 ± 0.00	27.51 ± 0.05	78.28 ± 0.15	89.13 ± 0.07	35.41 ± 0.06	24.08 ± 0.05	94.65 ± 0.14
BCA(Macro-P@3)	42.90 ± 0.01	43.70 ± 0.02	33.35 ± 0.01	**72.81** ± 0.20	11.89 ± 0.08	55.94 ± 0.04	14.13 ± 0.09	8.91 ± 0.06	88.34 ± 0.25
BCA(Macro-R@3)	42.11 ± 0.01	58.72 ± 0.02	31.69 ± 0.00	26.31 ± 0.03	**78.75** ± 0.13	**89.37** ± 0.07	33.12 ± 0.04	22.61 ± 0.03	*94.99* ± *0.16*
BCA(Macro-BA@3)	42.48 ± 0.01	59.09 ± 0.03	32.15 ± 0.00	26.33 ± 0.03	**78.75** ± 0.13	**89.37** ± 0.07	33.14 ± 0.04	22.62 ± 0.03	*94.99* ± *0.16*
BCA(Macro-F_1@3)	67.36 ± 0.00	83.70 ± 0.04	53.32 ± 0.01	53.07 ± 0.09	64.43 ± 0.08	82.21 ± 0.04	**56.50** ± 0.09	*42.34* ± *0.09*	93.40 ± 0.06
BCA(Macro-JS@3)	67.26 ± 0.01	83.63 ± 0.03	53.25 ± 0.01	52.96 ± 0.09	64.49 ± 0.08	82.24 ± 0.04	*56.48* ± *0.09*	**42.35** ± 0.09	93.39 ± 0.06
BCA(Cov@3)	2.42 ± 0.00	5.84 ± 0.01	1.36 ± 0.00	11.89 ± 0.04	42.34 ± 0.14	71.16 ± 0.07	9.27 ± 0.04	5.24 ± 0.03	**95.15** ± 0.14
FW(Macro-P@3)	40.87 ± 0.01	41.67 ± 0.01	31.51 ± 0.01	*71.62* ± *0.19*	11.83 ± 0.07	55.91 ± 0.03	13.86 ± 0.10	8.76 ± 0.07	86.39 ± 0.24
FW(Macro-R@3)	42.23 ± 0.01	58.86 ± 0.02	31.81 ± 0.00	26.25 ± 0.04	*78.72* ± *0.12*	*89.35* ± *0.06*	33.05 ± 0.05	22.55 ± 0.04	94.97 ± 0.16
FW(Macro-BA@3)	42.60 ± 0.01	59.23 ± 0.03	32.25 ± 0.00	26.28 ± 0.04	*78.72* ± *0.12*	*89.35* ± *0.06*	33.07 ± 0.05	22.57 ± 0.04	94.97 ± 0.16
FW(Macro-F_1@3)	67.34 ± 0.01	83.69 ± 0.03	53.29 ± 0.01	53.09 ± 0.09	64.30 ± 0.08	82.15 ± 0.04	56.44 ± 0.09	42.31 ± 0.08	93.26 ± 0.07
FW(Macro-JS@3)	67.25 ± 0.01	83.60 ± 0.04	53.23 ± 0.01	52.98 ± 0.09	64.36 ± 0.08	82.17 ± 0.04	56.42 ± 0.09	42.32 ± 0.08	93.24 ± 0.06

Method	Instance @3 P	Instance @3 PS	Instance @3 R	Macro @3 P	Macro @3 R	Macro @3 BA	Macro @3 F_1	Macro @3 JS	Cov
Synthetic AmazonCat-14K									
Top-3	**66.89 ± 0.00**	73.87 ± 0.01	* **68.62 ± 0.01**	40.94 ± 0.08	22.07 ± 0.05	61.03 ± 0.03	25.15 ± 0.04	16.88 ± 0.04	68.28 ± 0.13
PS-3	*65.57 ± 0.01*	**77.36 ± 0.01**	*67.56 ± 0.01*	41.88 ± 0.02	58.71 ± 0.10	79.35 ± 0.05	45.33 ± 0.03	32.46 ± 0.02	91.74 ± 0.11
Pow-3$_{\beta=0.25}$	61.37 ± 0.00	74.91 ± 0.01	63.90 ± 0.00	38.06 ± 0.04	63.87 ± 0.07	81.93 ± 0.03	45.23 ± 0.02	32.28 ± 0.03	91.68 ± 0.08
Pow-3$_{\beta=0.5}$	54.85 ± 0.01	69.02 ± 0.01	57.33 ± 0.01	29.65 ± 0.04	70.87 ± 0.08	85.43 ± 0.04	38.64 ± 0.04	26.64 ± 0.04	92.78 ± 0.10
Log-3	62.78 ± 0.00	*75.62 ± 0.01*	65.09 ± 0.00	42.08 ± 0.09	54.08 ± 0.09	77.04 ± 0.05	45.34 ± 0.06	32.76 ± 0.06	87.67 ± 0.09
Macro-R$_{\text{prior}}$-3	37.07 ± 0.00	50.67 ± 0.01	33.16 ± 0.00	20.45 ± 0.03	75.22 ± 0.13	87.60 ± 0.06	27.14 ± 0.04	17.99 ± 0.03	93.31 ± 0.12
Macro-BA$_{\text{prior}}$-3	37.39 ± 0.00	51.00 ± 0.01	33.96 ± 0.00	20.46 ± 0.02	75.22 ± 0.12	87.60 ± 0.06	27.16 ± 0.04	18.01 ± 0.03	93.31 ± 0.12
BCA(Macro-P@3)	34.24 ± 0.01	35.16 ± 0.01	40.07 ± 0.01	**72.26 ± 0.22**	7.91 ± 0.13	53.95 ± 0.06	10.81 ± 0.12	6.93 ± 0.11	82.16 ± 0.23
BCA(Macro-R@3)	35.10 ± 0.01	48.36 ± 0.01	30.35 ± 0.00	20.18 ± 0.03	**75.84 ± 0.08**	**87.91 ± 0.04**	26.11 ± 0.03	17.43 ± 0.03	*93.79 ± 0.07*
BCA(Macro-BA@3)	35.58 ± 0.01	48.85 ± 0.01	31.38 ± 0.00	20.20 ± 0.03	**75.84 ± 0.07**	**87.91 ± 0.04**	26.15 ± 0.03	17.45 ± 0.03	*93.79 ± 0.07*
BCA(Macro-F_1@3)	60.74 ± 0.00	73.34 ± 0.01	63.23 ± 0.00	48.83 ± 0.06	56.02 ± 0.14	78.01 ± 0.07	**51.01 ± 0.08**	**37.35 ± 0.08**	91.13 ± 0.07
BCA(Macro-JS@3)	60.70 ± 0.00	73.31 ± 0.01	63.18 ± 0.00	48.78 ± 0.06	56.05 ± 0.14	78.02 ± 0.07	*50.99 ± 0.08*	*37.34 ± 0.08*	91.12 ± 0.08
BCA(Cov@3)	1.60 ± 0.00	4.07 ± 0.01	1.21 ± 0.00	11.04 ± 0.01	40.20 ± 0.12	70.09 ± 0.06	5.96 ± 0.02	3.28 ± 0.02	**93.99 ± 0.08**
FW(Macro-P@3)	33.87 ± 0.01	34.80 ± 0.01	39.58 ± 0.01	*65.53 ± 0.18*	7.78 ± 0.09	53.88 ± 0.05	9.96 ± 0.08	6.36 ± 0.08	74.63 ± 0.17
FW(Macro-R@3)	35.38 ± 0.01	48.65 ± 0.01	31.08 ± 0.00	20.13 ± 0.03	75.73 ± 0.10	*87.86 ± 0.05*	26.05 ± 0.03	17.38 ± 0.03	93.71 ± 0.07
FW(Macro-BA@3)	35.26 ± 0.01	48.52 ± 0.01	30.80 ± 0.00	20.13 ± 0.03	*75.74 ± 0.10*	*87.86 ± 0.05*	26.04 ± 0.03	17.38 ± 0.03	93.72 ± 0.07
FW(Macro-F_1@3)	60.79 ± 0.00	73.39 ± 0.00	63.30 ± 0.00	48.74 ± 0.06	55.73 ± 0.11	77.86 ± 0.06	50.74 ± 0.07	37.14 ± 0.06	90.73 ± 0.06
FW(Macro-JS@3)	60.72 ± 0.00	73.33 ± 0.01	63.22 ± 0.00	48.71 ± 0.07	55.76 ± 0.13	77.87 ± 0.06	50.74 ± 0.08	37.15 ± 0.07	90.72 ± 0.07
Synthetic Wiki10-31K									
Top-3	**86.55 ± 0.08**	103.74 ± 1.04	* **24.55 ± 0.02**	5.61 ± 0.03	1.25 ± 0.01	50.62 ± 0.00	1.83 ± 0.01	1.17 ± 0.01	6.76 ± 0.01
PS-3	*67.31 ± 0.11*	**283.40 ± 0.57**	*19.80 ± 0.02*	51.82 ± 0.13	52.03 ± 0.08	76.01 ± 0.04	45.98 ± 0.10	36.75 ± 0.09	76.36 ± 0.11
Pow-3$_{\beta=0.25}$	65.95 ± 0.10	*281.88 ± 0.47*	19.05 ± 0.03	51.47 ± 0.09	50.97 ± 0.08	75.49 ± 0.04	45.55 ± 0.08	36.37 ± 0.08	75.65 ± 0.12
Pow-3$_{\beta=0.5}$	58.78 ± 0.12	277.87 ± 0.59	16.11 ± 0.04	49.94 ± 0.15	52.82 ± 0.08	76.41 ± 0.04	45.67 ± 0.11	36.28 ± 0.11	77.51 ± 0.12
Log-3	65.32 ± 0.08	271.23 ± 0.44	17.89 ± 0.03	48.98 ± 0.09	46.11 ± 0.08	73.05 ± 0.04	42.49 ± 0.07	33.76 ± 0.08	70.72 ± 0.10
Macro-R$_{\text{prior}}$-3	49.23 ± 0.10	265.34 ± 0.52	12.23 ± 0.03	45.26 ± 0.15	53.27 ± 0.10	76.63 ± 0.05	43.34 ± 0.11	33.89 ± 0.10	77.29 ± 0.14
Macro-BA$_{\text{prior}}$-3	49.33 ± 0.10	265.47 ± 0.52	12.28 ± 0.03	45.30 ± 0.15	53.27 ± 0.10	76.63 ± 0.05	43.34 ± 0.11	33.90 ± 0.10	77.29 ± 0.14
BCA(Macro-P@3)	63.21 ± 0.09	181.94 ± 0.45	17.64 ± 0.03	**64.76 ± 0.18**	35.44 ± 0.06	67.72 ± 0.03	38.77 ± 0.10	29.28 ± 0.09	81.09 ± 0.18
BCA(Macro-R@3)	47.81 ± 0.11	254.52 ± 0.65	11.69 ± 0.02	46.72 ± 0.14	**56.86 ± 0.14**	**78.43 ± 0.07**	45.77 ± 0.14	35.68 ± 0.13	82.24 ± 0.10
BCA(Macro-BA@3)	47.86 ± 0.11	254.57 ± 0.65	11.72 ± 0.02	46.73 ± 0.14	*56.85 ± 0.14*	*78.42 ± 0.07*	45.77 ± 0.14	35.68 ± 0.13	82.24 ± 0.09
BCA(Macro-F_1@3)	56.53 ± 0.13	261.00 ± 0.64	14.95 ± 0.03	57.99 ± 0.17	54.03 ± 0.11	77.01 ± 0.05	**51.41 ± 0.14**	*40.16 ± 0.14*	*89.93 ± 0.12*
BCA(Macro-JS@3)	55.69 ± 0.10	263.86 ± 0.47	14.63 ± 0.03	56.42 ± 0.11	55.33 ± 0.10	77.66 ± 0.05	*51.02 ± 0.10*	**40.29 ± 0.11**	87.94 ± 0.06
BCA(Cov@3)	46.04 ± 0.09	239.67 ± 0.48	10.97 ± 0.02	51.31 ± 0.11	53.63 ± 0.10	76.81 ± 0.05	46.72 ± 0.10	35.34 ± 0.09	**91.01 ± 0.16**
FW(Macro-P@3)	62.69 ± 0.06	180.63 ± 0.47	17.48 ± 0.02	*58.01 ± 0.16*	34.34 ± 0.09	67.17 ± 0.04	36.23 ± 0.11	27.59 ± 0.11	73.85 ± 0.18
FW(Macro-R@3)	47.82 ± 0.10	259.29 ± 0.55	11.69 ± 0.03	46.35 ± 0.17	54.60 ± 0.14	77.30 ± 0.07	44.34 ± 0.12	34.59 ± 0.11	79.98 ± 0.12
FW(Macro-BA@3)	47.87 ± 0.10	259.35 ± 0.56	11.72 ± 0.03	46.37 ± 0.17	54.60 ± 0.14	77.30 ± 0.07	44.34 ± 0.12	34.59 ± 0.11	79.99 ± 0.12
FW(Macro-F_1@3)	56.63 ± 0.14	266.42 ± 0.62	14.97 ± 0.04	56.51 ± 0.17	51.59 ± 0.09	75.79 ± 0.05	48.53 ± 0.13	38.02 ± 0.13	85.01 ± 0.16
FW(Macro-JS@3)	55.98 ± 0.13	268.86 ± 0.68	14.74 ± 0.03	55.50 ± 0.19	53.02 ± 0.16	76.51 ± 0.08	48.72 ± 0.16	38.52 ± 0.14	84.10 ± 0.15

Method	Instance @3			Macro @3					
	P	PS	R	P	R	BA	F_1	JS	Cov
Synthetic WikiLSHTC-325K									
Top-3	**41.99** ± **0.01**	90.10 ± 0.01	* **58.69** ± **0.01**	30.18 ± 0.03	47.83 ± 0.04	73.92 ± 0.02	33.81 ± 0.03	23.86 ± 0.02	71.06 ± 0.06
PS-3	*40.39* ± *0.01*	**95.85** ± **0.02**	*57.36* ± *0.01*	32.57 ± 0.04	63.58 ± 0.05	81.79 ± 0.02	40.39 ± 0.04	28.75 ± 0.04	82.82 ± 0.06
Pow-$3_{\beta=0.25}$	39.13 ± 0.01	*95.09* ± *0.02*	55.88 ± 0.01	32.50 ± 0.04	62.81 ± 0.04	81.40 ± 0.02	40.33 ± 0.04	28.74 ± 0.03	82.14 ± 0.06
Pow-$3_{\beta=0.5}$	36.54 ± 0.01	93.06 ± 0.02	52.46 ± 0.01	30.00 ± 0.04	65.45 ± 0.06	82.72 ± 0.03	38.58 ± 0.05	27.10 ± 0.04	83.41 ± 0.07
Log-3	40.16 ± 0.01	93.95 ± 0.03	56.87 ± 0.01	32.31 ± 0.04	56.47 ± 0.04	78.24 ± 0.02	38.39 ± 0.04	27.35 ± 0.03	77.91 ± 0.06
Macro-R_{prior}-3	32.98 ± 0.01	88.68 ± 0.03	48.03 ± 0.01	25.92 ± 0.04	66.68 ± 0.07	83.34 ± 0.03	34.38 ± 0.04	23.57 ± 0.04	83.83 ± 0.08
Macro-BA_{prior}-3	32.98 ± 0.01	88.68 ± 0.03	48.03 ± 0.01	25.92 ± 0.04	66.68 ± 0.07	83.34 ± 0.03	34.38 ± 0.04	23.57 ± 0.04	83.83 ± 0.08
BCA(Macro-P@3)	17.10 ± 0.00	32.22 ± 0.05	21.84 ± 0.00	**50.46** ± **0.06**	29.16 ± 0.06	64.58 ± 0.03	27.43 ± 0.06	19.49 ± 0.05	75.42 ± 0.08
BCA(Macro-R@3)	29.72 ± 0.01	83.23 ± 0.03	43.72 ± 0.01	22.98 ± 0.02	**67.36** ± **0.05**	**83.68** ± **0.03**	30.34 ± 0.02	20.35 ± 0.02	*84.48* ± *0.07*
BCA(Macro-BA@3)	29.72 ± 0.01	83.23 ± 0.03	43.73 ± 0.01	22.98 ± 0.02	**67.36** ± **0.05**	**83.68** ± **0.03**	30.35 ± 0.02	20.35 ± 0.02	*84.48* ± *0.07*
BCA(Macro-F_1@3)	36.55 ± 0.01	87.96 ± 0.01	51.78 ± 0.00	41.10 ± 0.04	59.61 ± 0.05	79.80 ± 0.02	**46.34** ± **0.04**	**34.17** ± **0.03**	82.79 ± 0.07
BCA(Macro-JS@3)	36.55 ± 0.01	88.02 ± 0.01	51.75 ± 0.01	41.04 ± 0.04	59.61 ± 0.05	79.80 ± 0.02	*46.32* ± *0.04*	**34.17** ± **0.03**	82.74 ± 0.07
BCA(Cov@3)	7.88 ± 0.00	32.74 ± 0.02	10.54 ± 0.00	22.41 ± 0.01	53.00 ± 0.06	76.50 ± 0.03	19.99 ± 0.01	12.73 ± 0.01	**84.98** ± **0.08**
FW(Macro-P@3)	17.19 ± 0.00	32.38 ± 0.02	21.94 ± 0.00	*49.18* ± *0.06*	28.89 ± 0.06	64.45 ± 0.03	26.71 ± 0.06	19.01 ± 0.05	73.05 ± 0.08
FW(Macro-R@3)	29.72 ± 0.01	83.30 ± 0.03	43.73 ± 0.01	22.87 ± 0.02	*67.18* ± *0.06*	*83.59* ± *0.03*	30.06 ± 0.02	20.20 ± 0.02	84.31 ± 0.08
FW(Macro-BA@3)	29.72 ± 0.01	83.30 ± 0.03	43.73 ± 0.01	22.87 ± 0.02	*67.18* ± *0.06*	*83.59* ± *0.03*	30.06 ± 0.02	20.20 ± 0.02	84.31 ± 0.08
FW(Macro-F_1@3)	36.56 ± 0.01	87.94 ± 0.01	51.80 ± 0.00	41.04 ± 0.04	59.17 ± 0.05	79.59 ± 0.02	46.04 ± 0.04	*33.95* ± *0.03*	82.39 ± 0.07
FW(Macro-JS@3)	36.54 ± 0.01	87.98 ± 0.02	51.74 ± 0.00	40.99 ± 0.04	59.19 ± 0.05	79.59 ± 0.02	46.01 ± 0.04	*33.95* ± *0.03*	82.35 ± 0.07
Synthetic WikipediaLarge-500K									
Top-3	**51.86** ± **0.01**	117.56 ± 0.04	* **55.88** ± **0.01**	35.17 ± 0.03	40.49 ± 0.01	70.24 ± 0.01	33.52 ± 0.02	23.78 ± 0.02	70.34 ± 0.01
PS-3	48.35 ± 0.01	**134.46** ± **0.04**	53.61 ± 0.01	43.95 ± 0.03	67.25 ± 0.01	83.63 ± 0.00	49.62 ± 0.02	36.28 ± 0.02	92.53 ± 0.01
Pow-$3_{\beta=0.25}$	47.71 ± 0.01	*133.88* ± *0.05*	52.89 ± 0.01	44.09 ± 0.03	65.72 ± 0.01	82.86 ± 0.01	49.43 ± 0.03	36.21 ± 0.03	91.57 ± 0.01
Pow-$3_{\beta=0.5}$	45.10 ± 0.01	132.27 ± 0.04	50.15 ± 0.01	41.32 ± 0.02	69.14 ± 0.01	84.57 ± 0.00	48.29 ± 0.02	34.89 ± 0.02	93.12 ± 0.00
Log-3	*49.53* ± *0.01*	130.27 ± 0.04	*54.18* ± *0.01*	42.82 ± 0.03	55.30 ± 0.02	77.65 ± 0.01	44.60 ± 0.03	32.44 ± 0.03	84.63 ± 0.02
Macro-R_{prior}-3	40.94 ± 0.02	127.00 ± 0.05	45.51 ± 0.01	36.89 ± 0.02	70.44 ± 0.01	85.22 ± 0.00	44.54 ± 0.02	31.37 ± 0.02	93.55 ± 0.00
Macro-BA_{prior}-3	40.95 ± 0.02	127.00 ± 0.05	45.51 ± 0.01	36.89 ± 0.02	70.44 ± 0.01	85.22 ± 0.00	44.54 ± 0.02	31.37 ± 0.02	93.55 ± 0.00
BCA(Macro-P@3)	24.07 ± 0.00	49.61 ± 0.01	22.34 ± 0.00	**62.98** ± **0.02**	28.78 ± 0.01	64.39 ± 0.01	30.96 ± 0.02	21.64 ± 0.02	87.39 ± 0.03
BCA(Macro-R@3)	37.83 ± 0.01	121.04 ± 0.04	41.90 ± 0.01	33.78 ± 0.02	**71.57** ± **0.01**	**85.78** ± **0.01**	40.80 ± 0.01	28.09 ± 0.01	*94.34* ± *0.00*
BCA(Macro-BA@3)	37.83 ± 0.01	121.04 ± 0.04	41.90 ± 0.01	33.78 ± 0.02	**71.57** ± **0.01**	**85.78** ± **0.01**	40.80 ± 0.01	28.09 ± 0.01	*94.34* ± *0.00*
BCA(Macro-F_1@3)	44.78 ± 0.01	126.54 ± 0.03	49.13 ± 0.01	52.56 ± 0.03	65.40 ± 0.00	82.70 ± 0.00	**55.52** ± **0.02**	*41.78* ± *0.02*	93.29 ± 0.01
BCA(Macro-JS@3)	44.74 ± 0.01	126.73 ± 0.04	49.07 ± 0.01	52.37 ± 0.03	65.50 ± 0.00	82.75 ± 0.00	*55.45* ± *0.02*	**41.80** ± **0.02**	93.21 ± 0.01
BCA(Cov@3)	14.26 ± 0.00	59.47 ± 0.02	14.11 ± 0.01	30.33 ± 0.01	56.29 ± 0.02	78.15 ± 0.01	26.93 ± 0.01	17.35 ± 0.01	**94.88** ± **0.00**
FW(Macro-P@3)	24.25 ± 0.01	49.90 ± 0.01	22.35 ± 0.01	*61.26* ± *0.03*	28.36 ± 0.01	64.18 ± 0.01	29.93 ± 0.02	21.00 ± 0.02	84.10 ± 0.03
FW(Macro-R@3)	37.83 ± 0.01	121.18 ± 0.04	41.89 ± 0.01	33.78 ± 0.02	*71.27* ± *0.01*	*85.64* ± *0.01*	40.69 ± 0.02	28.01 ± 0.02	94.18 ± 0.01
FW(Macro-BA@3)	37.83 ± 0.01	121.18 ± 0.04	41.89 ± 0.01	33.78 ± 0.02	*71.27* ± *0.01*	*85.64* ± *0.01*	40.69 ± 0.02	28.01 ± 0.02	94.18 ± 0.01
FW(Macro-F_1@3)	44.75 ± 0.01	126.57 ± 0.04	49.10 ± 0.01	52.55 ± 0.03	64.92 ± 0.01	82.46 ± 0.00	55.16 ± 0.02	41.52 ± 0.02	92.91 ± 0.01
FW(Macro-JS@3)	44.73 ± 0.01	126.75 ± 0.04	49.05 ± 0.01	52.36 ± 0.03	65.04 ± 0.01	82.52 ± 0.00	55.12 ± 0.02	41.56 ± 0.03	92.84 ± 0.01

Method	Instance @3 P	Instance @3 PS	Instance @3 R	Macro @3 P	Macro @3 R	Macro @3 BA	Macro @3 F_1	Macro @3 JS	Macro @3 Cov
Synthetic Amazon-670K									
Top-3	**47.56 ± 0.01**	269.66 ± 0.06	* **44.35 ± 0.01**	36.63 ± 0.01	43.27 ± 0.01	71.63 ± 0.01	36.29 ± 0.01	26.76 ± 0.01	67.89 ± 0.02
PS-3	44.98 ± 0.01	**302.09 ± 0.10**	41.99 ± 0.01	41.04 ± 0.02	55.43 ± 0.02	77.71 ± 0.01	42.78 ± 0.02	31.73 ± 0.02	81.01 ± 0.02
Pow-3$_{\beta=0.25}$	45.91 ± 0.01	300.64 ± 0.09	42.87 ± 0.01	41.48 ± 0.02	53.73 ± 0.02	76.87 ± 0.01	42.63 ± 0.02	31.68 ± 0.02	79.87 ± 0.02
Pow-3$_{\beta=0.5}$	44.48 ± 0.01	*301.64 ± 0.11*	41.49 ± 0.01	40.67 ± 0.02	55.89 ± 0.02	77.94 ± 0.01	42.56 ± 0.02	31.51 ± 0.02	81.21 ± 0.02
Log-3	*47.13 ± 0.01*	289.36 ± 0.05	*43.96 ± 0.01*	40.00 ± 0.01	48.78 ± 0.01	74.39 ± 0.00	40.18 ± 0.01	29.80 ± 0.01	74.83 ± 0.01
Macro-R$_{\text{prior}}$-3	42.83 ± 0.02	298.31 ± 0.13	39.76 ± 0.02	39.12 ± 0.02	56.61 ± 0.02	78.31 ± 0.01	41.32 ± 0.02	30.38 ± 0.02	80.77 ± 0.02
Macro-BA$_{\text{prior}}$-3	42.83 ± 0.02	298.31 ± 0.13	39.76 ± 0.02	39.12 ± 0.02	56.61 ± 0.02	78.31 ± 0.01	41.32 ± 0.02	30.38 ± 0.02	80.77 ± 0.02
BCA(Macro-P@3)	27.71 ± 0.00	170.25 ± 0.04	23.55 ± 0.01	**56.57 ± 0.02**	34.39 ± 0.02	67.19 ± 0.01	36.47 ± 0.01	26.25 ± 0.01	79.40 ± 0.02
BCA(Macro-R@3)	41.82 ± 0.02	289.97 ± 0.13	38.72 ± 0.02	39.07 ± 0.02	**58.38 ± 0.03**	**79.19 ± 0.01**	41.56 ± 0.02	30.31 ± 0.02	83.18 ± 0.03
BCA(Macro-BA@3)	41.82 ± 0.02	289.97 ± 0.13	38.72 ± 0.02	39.07 ± 0.02	**58.38 ± 0.03**	**79.19 ± 0.01**	41.56 ± 0.02	30.31 ± 0.02	83.18 ± 0.03
BCA(Macro-F_1@3)	44.09 ± 0.02	286.09 ± 0.13	40.45 ± 0.01	50.22 ± 0.03	54.29 ± 0.02	77.14 ± 0.01	**48.60 ± 0.02**	*35.98 ± 0.02*	*89.34 ± 0.02*
BCA(Macro-JS@3)	44.07 ± 0.02	288.92 ± 0.14	40.48 ± 0.02	49.29 ± 0.03	54.94 ± 0.02	77.47 ± 0.01	*48.34 ± 0.02*	**36.03 ± 0.02**	88.56 ± 0.02
BCA(Cov@3)	34.19 ± 0.02	253.01 ± 0.18	32.65 ± 0.02	43.53 ± 0.02	54.47 ± 0.02	77.23 ± 0.01	41.40 ± 0.03	29.22 ± 0.02	**91.16 ± 0.01**
FW(Macro-P@3)	26.90 ± 0.01	169.68 ± 0.04	22.79 ± 0.01	*52.24 ± 0.02*	33.67 ± 0.02	66.84 ± 0.01	34.52 ± 0.01	24.95 ± 0.01	74.38 ± 0.02
FW(Macro-R@3)	41.82 ± 0.02	292.31 ± 0.15	38.72 ± 0.02	39.18 ± 0.03	*57.49 ± 0.02*	*78.74 ± 0.01*	41.13 ± 0.03	30.08 ± 0.02	81.96 ± 0.02
FW(Macro-BA@3)	41.83 ± 0.02	292.32 ± 0.15	38.72 ± 0.02	39.18 ± 0.03	*57.49 ± 0.02*	*78.74 ± 0.01*	41.13 ± 0.03	30.08 ± 0.02	81.96 ± 0.02
FW(Macro-F_1@3)	43.99 ± 0.02	288.38 ± 0.14	40.33 ± 0.01	49.57 ± 0.03	53.10 ± 0.02	76.55 ± 0.01	46.92 ± 0.02	34.82 ± 0.02	86.52 ± 0.02
FW(Macro-JS@3)	43.98 ± 0.01	290.88 ± 0.12	40.35 ± 0.01	48.96 ± 0.03	53.84 ± 0.01	76.92 ± 0.01	46.96 ± 0.02	35.09 ± 0.02	86.01 ± 0.02

Method	Instance @5 P	Instance @5 PS	Instance @5 R	Macro @5 P	Macro @5 R	Macro @5 BA	Macro @5 F_1	Macro @5 JS	Macro @5 Cov
Synthetic RCV1x-2K									
Top-5	**54.70 ± 0.01**	*57.18 ± 0.12*	* **76.75 ± 0.02**	32.50 ± 0.27	17.05 ± 0.05	58.48 ± 0.02	17.35 ± 0.07	10.78 ± 0.04	76.95 ± 0.09
PS-5	*54.32 ± 0.01*	**57.88 ± 0.11**	*76.55 ± 0.02*	38.15 ± 0.06	45.91 ± 0.10	72.91 ± 0.05	36.11 ± 0.07	23.32 ± 0.05	99.70 ± 0.07
Pow-5$_{\beta=0.25}$	52.66 ± 0.01	56.90 ± 0.05	75.37 ± 0.02	33.83 ± 0.06	53.57 ± 0.11	76.74 ± 0.06	37.38 ± 0.07	24.25 ± 0.05	99.70 ± 0.06
Pow-5$_{\beta=0.5}$	49.42 ± 0.01	54.08 ± 0.04	72.42 ± 0.02	25.39 ± 0.05	62.15 ± 0.15	81.02 ± 0.07	31.06 ± 0.04	19.74 ± 0.03	*99.84 ± 0.06*
Log-5	52.99 ± 0.01	57.10 ± 0.04	75.52 ± 0.02	37.81 ± 0.11	44.77 ± 0.12	72.34 ± 0.06	37.00 ± 0.08	23.92 ± 0.06	98.94 ± 0.09
Macro-R$_{\text{prior}}$-5	36.69 ± 0.01	41.37 ± 0.06	56.73 ± 0.01	17.31 ± 0.02	68.59 ± 0.14	84.23 ± 0.07	19.86 ± 0.02	12.20 ± 0.01	**99.90 ± 0.06**
Macro-BA$_{\text{prior}}$-5	37.13 ± 0.01	41.79 ± 0.05	57.43 ± 0.01	17.36 ± 0.02	68.58 ± 0.14	84.23 ± 0.07	19.92 ± 0.02	12.24 ± 0.01	**99.90 ± 0.06**
BCA(Macro-P@5)	32.75 ± 0.01	33.19 ± 0.05	49.34 ± 0.01	**74.72 ± 0.58**	5.46 ± 0.13	52.68 ± 0.06	7.57 ± 0.14	4.50 ± 0.09	92.61 ± 0.49
BCA(Macro-R@5)	31.98 ± 0.01	36.47 ± 0.03	49.79 ± 0.01	16.98 ± 0.02	**69.30 ± 0.11**	**84.58 ± 0.06**	18.40 ± 0.02	11.28 ± 0.01	**99.90 ± 0.06**
BCA(Macro-BA@5)	32.66 ± 0.01	37.13 ± 0.02	50.92 ± 0.01	17.02 ± 0.02	**69.30 ± 0.11**	**84.58 ± 0.06**	18.46 ± 0.02	11.33 ± 0.01	**99.90 ± 0.06**
BCA(Macro-F_1@5)	49.19 ± 0.01	53.35 ± 0.07	71.05 ± 0.01	46.34 ± 0.07	50.10 ± 0.13	75.00 ± 0.06	**46.56 ± 0.06**	*31.35 ± 0.04*	99.69 ± 0.12
BCA(Macro-JS@5)	49.24 ± 0.01	53.40 ± 0.04	71.02 ± 0.01	46.07 ± 0.07	50.14 ± 0.12	75.02 ± 0.06	*46.47 ± 0.05*	**31.36 ± 0.04**	99.68 ± 0.13
BCA(Cov@5)	1.14 ± 0.00	2.22 ± 0.00	0.77 ± 0.00	11.12 ± 0.02	44.85 ± 0.11	72.32 ± 0.05	4.78 ± 0.02	2.54 ± 0.01	**99.90 ± 0.06**
FW(Macro-P@5)	30.63 ± 0.01	31.08 ± 0.05	46.21 ± 0.01	*72.33 ± 0.53*	6.22 ± 0.12	53.05 ± 0.06	7.36 ± 0.14	4.37 ± 0.09	89.88 ± 0.51
FW(Macro-R@5)	31.93 ± 0.01	36.42 ± 0.03	49.69 ± 0.01	16.98 ± 0.02	*69.29 ± 0.11*	*84.57 ± 0.05*	18.39 ± 0.02	11.28 ± 0.01	**99.90 ± 0.06**
FW(Macro-BA@5)	32.62 ± 0.01	37.09 ± 0.02	50.85 ± 0.01	17.02 ± 0.02	69.28 ± 0.11	*84.57 ± 0.05*	18.45 ± 0.02	11.32 ± 0.01	**99.90 ± 0.06**
FW(Macro-F_1@5)	49.17 ± 0.01	53.31 ± 0.08	71.06 ± 0.02	46.11 ± 0.07	49.85 ± 0.12	74.88 ± 0.06	46.24 ± 0.07	31.14 ± 0.05	99.57 ± 0.17
FW(Macro-JS@5)	49.22 ± 0.01	53.37 ± 0.08	71.00 ± 0.01	45.89 ± 0.07	49.88 ± 0.14	74.89 ± 0.07	46.21 ± 0.07	31.19 ± 0.06	99.54 ± 0.17

Method	Instance @5			Macro @5					
	P	PS	R	P	R	BA	F_1	JS	Cov
Synthetic EURLex-4K									
Top-5	**65.85** ± 0.03	130.35 ± 0.07	* **73.84** ± 0.05	51.16 ± 0.17	57.70 ± 0.11	78.83 ± 0.05	52.42 ± 0.09	40.40 ± 0.09	82.77 ± 0.13
PS-5	64.83 ± 0.03	**134.08** ± 0.09	72.77 ± 0.06	49.35 ± 0.07	67.84 ± 0.13	83.90 ± 0.06	55.17 ± 0.06	42.29 ± 0.05	88.41 ± 0.09
Pow-5$_{\beta=0.25}$	64.74 ± 0.03	*133.95* ± *0.10*	72.67 ± 0.06	50.22 ± 0.06	67.31 ± 0.13	83.63 ± 0.06	55.66 ± 0.03	42.82 ± 0.02	88.15 ± 0.10
Pow-5$_{\beta=0.5}$	62.14 ± 0.04	132.08 ± 0.13	69.81 ± 0.07	45.55 ± 0.08	69.69 ± 0.08	84.82 ± 0.04	52.64 ± 0.04	39.72 ± 0.04	88.85 ± 0.12
Log-5	*65.05* ± *0.03*	133.80 ± 0.10	*73.02* ± *0.05*	51.46 ± 0.09	66.08 ± 0.16	83.02 ± 0.08	56.13 ± 0.05	43.38 ± 0.05	87.69 ± 0.11
Macro-R$_{\text{prior}}$-5	49.94 ± 0.05	116.20 ± 0.11	56.13 ± 0.06	34.98 ± 0.06	72.05 ± 0.19	85.99 ± 0.09	41.78 ± 0.05	29.80 ± 0.05	90.12 ± 0.13
Macro-BA$_{\text{prior}}$-5	50.06 ± 0.05	116.37 ± 0.11	56.29 ± 0.06	35.03 ± 0.06	72.04 ± 0.19	85.99 ± 0.09	41.84 ± 0.05	29.84 ± 0.05	90.12 ± 0.13
BCA(Macro-P@5)	20.47 ± 0.03	36.54 ± 0.12	22.48 ± 0.03	**67.71** ± 0.11	24.95 ± 0.31	62.44 ± 0.15	27.13 ± 0.15	19.45 ± 0.12	84.04 ± 0.24
BCA(Macro-R@5)	49.33 ± 0.04	115.16 ± 0.09	55.38 ± 0.04	34.11 ± 0.07	**72.83** ± 0.18	**86.38** ± 0.09	40.79 ± 0.09	28.96 ± 0.07	*90.70* ± *0.14*
BCA(Macro-BA@5)	49.46 ± 0.04	115.34 ± 0.09	55.53 ± 0.04	34.17 ± 0.07	*72.82* ± *0.19*	**86.38** ± 0.09	40.87 ± 0.08	29.03 ± 0.07	90.68 ± 0.16
BCA(Macro-F_1@5)	62.65 ± 0.03	129.11 ± 0.09	69.84 ± 0.04	56.06 ± 0.11	64.88 ± 0.16	82.42 ± 0.08	**58.51** ± 0.06	**45.85** ± 0.06	88.66 ± 0.16
BCA(Macro-JS@5)	62.66 ± 0.03	129.12 ± 0.10	69.81 ± 0.03	56.07 ± 0.12	64.86 ± 0.14	82.41 ± 0.07	*58.50* ± *0.06*	**45.85** ± 0.06	88.66 ± 0.16
BCA(Cov@5)	9.12 ± 0.03	34.62 ± 0.11	10.19 ± 0.03	28.67 ± 0.08	49.89 ± 0.11	74.89 ± 0.06	17.65 ± 0.06	10.85 ± 0.07	**91.39** ± 0.18
FW(Macro-P@5)	19.72 ± 0.04	35.46 ± 0.14	21.61 ± 0.05	*65.70* ± *0.11*	24.74 ± 0.29	62.33 ± 0.14	26.31 ± 0.17	18.86 ± 0.15	82.03 ± 0.12
FW(Macro-R@5)	48.42 ± 0.04	113.96 ± 0.10	54.32 ± 0.04	33.54 ± 0.05	72.47 ± 0.15	*86.20* ± *0.07*	40.05 ± 0.04	28.34 ± 0.04	90.44 ± 0.12
FW(Macro-BA@5)	48.54 ± 0.04	114.13 ± 0.10	54.46 ± 0.04	33.60 ± 0.04	72.47 ± 0.14	*86.20* ± *0.07*	40.11 ± 0.03	28.40 ± 0.03	90.48 ± 0.13
FW(Macro-F_1@5)	62.61 ± 0.04	129.04 ± 0.11	69.80 ± 0.04	55.94 ± 0.14	64.69 ± 0.14	82.32 ± 0.07	58.27 ± 0.08	*45.63* ± *0.07*	88.48 ± 0.15
FW(Macro-JS@5)	62.62 ± 0.04	129.07 ± 0.13	69.78 ± 0.03	55.94 ± 0.13	64.69 ± 0.15	82.32 ± 0.07	58.25 ± 0.08	*45.63* ± *0.07*	88.44 ± 0.12
Synthetic EURLex-4.3K									
Top-5	**73.79** ± 0.05	114.28 ± 0.18	* **82.02** ± 0.07	52.29 ± 0.21	57.28 ± 0.17	78.62 ± 0.08	52.71 ± 0.15	41.14 ± 0.14	81.34 ± 0.09
PS-5	*72.93* ± *0.07*	**118.07** ± 0.22	*81.21* ± *0.08*	53.49 ± 0.12	75.26 ± 0.21	87.61 ± 0.10	60.50 ± 0.11	47.67 ± 0.13	90.93 ± 0.18
Pow-5$_{\beta=0.25}$	72.53 ± 0.07	*117.97* ± *0.14*	80.83 ± 0.08	53.93 ± 0.10	75.15 ± 0.19	87.56 ± 0.10	60.91 ± 0.08	48.11 ± 0.10	90.80 ± 0.19
Pow-5$_{\beta=0.5}$	70.68 ± 0.07	116.64 ± 0.11	78.94 ± 0.07	49.28 ± 0.10	77.53 ± 0.16	88.75 ± 0.08	57.61 ± 0.09	44.74 ± 0.12	91.40 ± 0.17
Log-5	72.77 ± 0.07	117.81 ± 0.12	81.06 ± 0.08	55.25 ± 0.15	73.25 ± 0.22	86.61 ± 0.11	61.26 ± 0.12	48.61 ± 0.12	90.07 ± 0.17
Macro-R$_{\text{prior}}$-5	61.88 ± 0.06	106.33 ± 0.17	69.15 ± 0.05	38.49 ± 0.12	79.85 ± 0.17	89.90 ± 0.09	46.66 ± 0.12	34.52 ± 0.10	92.32 ± 0.21
Macro-BA$_{\text{prior}}$-5	62.04 ± 0.06	106.52 ± 0.18	69.33 ± 0.05	38.55 ± 0.12	79.85 ± 0.18	89.90 ± 0.09	46.72 ± 0.12	34.57 ± 0.10	92.32 ± 0.21
BCA(Macro-P@5)	30.93 ± 0.02	41.34 ± 0.07	33.80 ± 0.03	**70.96** ± 0.18	24.92 ± 0.13	62.43 ± 0.06	26.70 ± 0.06	19.23 ± 0.07	86.62 ± 0.16
BCA(Macro-R@5)	60.15 ± 0.07	104.09 ± 0.13	67.17 ± 0.05	37.05 ± 0.13	**80.19** ± 0.14	**90.07** ± 0.07	44.89 ± 0.13	32.99 ± 0.12	*92.60* ± *0.16*
BCA(Macro-BA@5)	60.32 ± 0.07	104.29 ± 0.14	67.38 ± 0.06	37.10 ± 0.13	*80.18* ± *0.14*	**90.07** ± 0.07	44.95 ± 0.13	33.04 ± 0.12	*92.60* ± *0.16*
BCA(Macro-F_1@5)	70.77 ± 0.06	114.91 ± 0.10	78.50 ± 0.07	60.70 ± 0.06	73.15 ± 0.18	86.56 ± 0.09	**64.76** ± 0.06	*52.36* ± *0.05*	91.10 ± 0.18
BCA(Macro-JS@5)	70.70 ± 0.06	114.84 ± 0.09	78.40 ± 0.07	60.69 ± 0.06	73.15 ± 0.17	86.56 ± 0.09	*64.75* ± *0.06*	**52.38** ± 0.05	91.10 ± 0.17
BCA(Cov@5)	5.83 ± 0.02	19.72 ± 0.08	6.28 ± 0.02	22.97 ± 0.04	53.56 ± 0.21	76.72 ± 0.10	16.62 ± 0.06	10.38 ± 0.04	**92.89** ± 0.22
FW(Macro-P@5)	30.84 ± 0.02	41.04 ± 0.09	33.69 ± 0.03	*68.89* ± *0.22*	24.64 ± 0.13	62.29 ± 0.06	26.06 ± 0.07	18.77 ± 0.07	84.32 ± 0.15
FW(Macro-R@5)	60.22 ± 0.06	104.17 ± 0.11	67.23 ± 0.05	36.91 ± 0.11	80.05 ± 0.18	*90.00* ± *0.09*	44.69 ± 0.12	32.84 ± 0.10	92.52 ± 0.18
FW(Macro-BA@5)	60.39 ± 0.06	104.40 ± 0.15	67.43 ± 0.05	36.96 ± 0.11	80.05 ± 0.18	*90.00* ± *0.09*	44.74 ± 0.12	32.89 ± 0.10	92.52 ± 0.18
FW(Macro-F_1@5)	70.76 ± 0.05	114.94 ± 0.10	78.50 ± 0.07	60.62 ± 0.07	72.96 ± 0.19	86.46 ± 0.10	64.60 ± 0.07	52.22 ± 0.05	90.92 ± 0.20
FW(Macro-JS@5)	70.69 ± 0.05	114.85 ± 0.08	78.40 ± 0.07	60.60 ± 0.07	72.98 ± 0.20	86.47 ± 0.10	64.57 ± 0.07	52.23 ± 0.05	90.95 ± 0.18

Method	Instance @5 P	Instance @5 PS	Instance @5 R	Macro @5 P	Macro @5 R	Macro @5 BA	Macro @5 F_1	Macro @5 JS	Macro @5 Cov
Synthetic AmazonCat-13K									
Top-5	**63.16 ± 0.01**	71.34 ± 0.17	* **76.12 ± 0.01**	45.39 ± 0.10	47.71 ± 0.06	73.85 ± 0.03	42.12 ± 0.06	30.42 ± 0.06	82.48 ± 0.11
PS-5	*62.63 ± 0.01*	**72.70 ± 0.14**	*75.72 ± 0.01*	43.95 ± 0.09	71.44 ± 0.15	85.71 ± 0.07	51.43 ± 0.10	37.33 ± 0.08	93.86 ± 0.13
Pow-5$_{\beta=0.25}$	59.86 ± 0.01	*71.35 ± 0.02*	73.51 ± 0.01	40.80 ± 0.09	75.01 ± 0.12	87.50 ± 0.06	50.70 ± 0.10	36.52 ± 0.08	93.79 ± 0.12
Pow-5$_{\beta=0.5}$	54.68 ± 0.01	66.80 ± 0.03	68.01 ± 0.01	31.75 ± 0.07	80.13 ± 0.18	90.06 ± 0.09	42.63 ± 0.08	29.53 ± 0.07	94.44 ± 0.15
Log-5	59.92 ± 0.01	71.18 ± 0.02	73.44 ± 0.01	44.13 ± 0.09	69.47 ± 0.07	84.73 ± 0.04	52.08 ± 0.09	37.99 ± 0.08	92.03 ± 0.04
Macro-R$_{\text{prior}}$-5	38.70 ± 0.01	50.58 ± 0.02	47.71 ± 0.01	20.81 ± 0.02	83.87 ± 0.12	91.92 ± 0.06	28.83 ± 0.03	19.13 ± 0.02	94.85 ± 0.14
Macro-BA$_{\text{prior}}$-5	39.08 ± 0.01	50.96 ± 0.02	48.34 ± 0.01	20.84 ± 0.02	83.87 ± 0.12	91.92 ± 0.06	28.86 ± 0.03	19.15 ± 0.02	94.85 ± 0.14
BCA(Macro-P@5)	33.90 ± 0.00	34.54 ± 0.11	42.20 ± 0.01	**72.72 ± 0.21**	12.27 ± 0.08	56.12 ± 0.04	14.55 ± 0.09	9.20 ± 0.06	88.37 ± 0.25
BCA(Macro-R@5)	35.31 ± 0.01	46.89 ± 0.01	43.42 ± 0.01	20.54 ± 0.03	**84.33 ± 0.13**	**92.15 ± 0.06**	27.82 ± 0.04	18.62 ± 0.03	*95.11 ± 0.14*
BCA(Macro-BA@5)	35.68 ± 0.01	47.25 ± 0.00	44.04 ± 0.00	20.56 ± 0.03	**84.33 ± 0.13**	**92.15 ± 0.06**	27.84 ± 0.04	18.64 ± 0.03	*95.11 ± 0.14*
BCA(Macro-F_1@5)	57.09 ± 0.00	67.75 ± 0.01	69.94 ± 0.01	52.89 ± 0.09	66.34 ± 0.09	83.16 ± 0.04	**57.45 ± 0.09**	**43.31 ± 0.09**	93.41 ± 0.06
BCA(Macro-JS@5)	56.98 ± 0.00	67.65 ± 0.03	69.76 ± 0.01	52.87 ± 0.09	66.35 ± 0.09	83.17 ± 0.04	*57.44 ± 0.09*	**43.31 ± 0.09**	93.41 ± 0.06
BCA(Cov@5)	2.62 ± 0.00	5.14 ± 0.01	2.43 ± 0.00	8.11 ± 0.03	47.63 ± 0.14	73.80 ± 0.07	7.87 ± 0.04	4.39 ± 0.02	**95.13 ± 0.09**
FW(Macro-P@5)	33.30 ± 0.00	33.92 ± 0.06	41.31 ± 0.01	*71.50 ± 0.20*	12.23 ± 0.07	56.10 ± 0.04	14.20 ± 0.10	9.00 ± 0.07	86.48 ± 0.26
FW(Macro-R@5)	35.37 ± 0.01	46.95 ± 0.02	43.50 ± 0.00	20.50 ± 0.03	*84.29 ± 0.13*	*92.13 ± 0.07*	27.76 ± 0.04	18.57 ± 0.03	95.08 ± 0.15
FW(Macro-BA@5)	35.74 ± 0.01	47.31 ± 0.01	44.13 ± 0.00	20.52 ± 0.03	*84.29 ± 0.13*	*92.13 ± 0.07*	27.78 ± 0.04	18.59 ± 0.03	95.08 ± 0.15
FW(Macro-F_1@5)	57.07 ± 0.00	67.75 ± 0.03	69.92 ± 0.01	52.90 ± 0.09	66.24 ± 0.09	83.11 ± 0.04	57.37 ± 0.10	*43.27 ± 0.09*	93.28 ± 0.08
FW(Macro-JS@5)	56.94 ± 0.00	67.62 ± 0.03	69.72 ± 0.01	52.86 ± 0.10	66.24 ± 0.09	83.11 ± 0.05	57.35 ± 0.10	43.26 ± 0.09	93.27 ± 0.08
Synthetic AmazonCat-14K									
Top-5	**52.43 ± 0.00**	60.15 ± 0.01	* **83.00 ± 0.01**	37.63 ± 0.04	50.29 ± 0.11	75.13 ± 0.05	40.68 ± 0.06	29.11 ± 0.05	81.49 ± 0.16
PS-5	*52.19 ± 0.00*	**60.67 ± 0.00**	*82.73 ± 0.01*	33.47 ± 0.03	67.54 ± 0.08	83.76 ± 0.04	42.04 ± 0.02	29.44 ± 0.03	92.44 ± 0.09
Pow-5$_{\beta=0.25}$	50.84 ± 0.00	59.90 ± 0.01	81.10 ± 0.01	30.73 ± 0.04	70.56 ± 0.09	85.27 ± 0.05	40.55 ± 0.03	28.03 ± 0.03	92.41 ± 0.09
Pow-5$_{\beta=0.5}$	45.94 ± 0.00	55.54 ± 0.00	74.21 ± 0.00	22.26 ± 0.03	76.95 ± 0.10	88.47 ± 0.05	31.81 ± 0.03	21.10 ± 0.03	93.18 ± 0.10
Log-5	51.45 ± 0.00	*60.22 ± 0.01*	81.85 ± 0.01	34.67 ± 0.04	63.84 ± 0.09	81.91 ± 0.05	43.04 ± 0.03	30.29 ± 0.03	90.11 ± 0.09
Macro-R$_{\text{prior}}$-5	29.69 ± 0.00	39.03 ± 0.01	44.01 ± 0.00	14.93 ± 0.02	81.32 ± 0.08	90.65 ± 0.04	21.33 ± 0.03	13.78 ± 0.02	93.63 ± 0.10
Macro-BA$_{\text{prior}}$-5	29.92 ± 0.00	39.26 ± 0.01	44.84 ± 0.00	14.94 ± 0.02	81.32 ± 0.08	90.65 ± 0.04	21.35 ± 0.03	13.79 ± 0.02	93.63 ± 0.10
BCA(Macro-P@5)	25.93 ± 0.00	26.70 ± 0.01	47.71 ± 0.01	**72.16 ± 0.22**	8.24 ± 0.11	54.11 ± 0.06	11.13 ± 0.11	7.15 ± 0.10	82.25 ± 0.23
BCA(Macro-R@5)	26.50 ± 0.00	35.59 ± 0.01	36.55 ± 0.01	14.90 ± 0.02	**81.95 ± 0.07**	**90.96 ± 0.04**	20.75 ± 0.03	13.52 ± 0.02	*94.00 ± 0.08*
BCA(Macro-BA@5)	29.48 ± 0.00	38.66 ± 0.01	43.27 ± 0.00	15.24 ± 0.02	*81.93 ± 0.07*	*90.95 ± 0.04*	21.28 ± 0.03	13.91 ± 0.02	*94.00 ± 0.08*
BCA(Macro-F_1@5)	48.73 ± 0.00	56.60 ± 0.00	77.80 ± 0.00	48.69 ± 0.05	57.31 ± 0.14	78.65 ± 0.07	**51.58 ± 0.08**	**37.86 ± 0.07**	91.22 ± 0.07
BCA(Macro-JS@5)	48.60 ± 0.00	56.44 ± 0.00	77.54 ± 0.01	48.74 ± 0.05	57.27 ± 0.14	78.63 ± 0.07	**51.58 ± 0.07**	**37.86 ± 0.07**	91.22 ± 0.07
BCA(Cov@5)	1.36 ± 0.00	3.29 ± 0.00	1.72 ± 0.00	7.76 ± 0.02	46.83 ± 0.11	73.40 ± 0.06	5.20 ± 0.02	2.84 ± 0.02	**94.08 ± 0.09**
FW(Macro-P@5)	25.83 ± 0.00	26.60 ± 0.01	47.52 ± 0.01	*65.54 ± 0.17*	8.15 ± 0.08	54.07 ± 0.04	10.28 ± 0.08	6.57 ± 0.08	74.90 ± 0.17
FW(Macro-R@5)	26.50 ± 0.00	35.58 ± 0.01	36.56 ± 0.01	14.86 ± 0.02	81.86 ± 0.10	90.92 ± 0.05	20.71 ± 0.03	13.49 ± 0.02	93.95 ± 0.09
FW(Macro-BA@5)	29.49 ± 0.00	38.66 ± 0.01	43.36 ± 0.00	15.20 ± 0.03	81.84 ± 0.10	90.91 ± 0.05	21.23 ± 0.03	13.88 ± 0.03	93.95 ± 0.09
FW(Macro-F_1@5)	48.72 ± 0.00	56.59 ± 0.00	77.79 ± 0.01	48.62 ± 0.06	57.02 ± 0.12	78.50 ± 0.06	*51.33 ± 0.07*	*37.67 ± 0.07*	90.85 ± 0.07
FW(Macro-JS@5)	48.62 ± 0.00	56.46 ± 0.00	77.58 ± 0.01	48.67 ± 0.07	56.98 ± 0.13	78.48 ± 0.06	51.32 ± 0.08	*37.67 ± 0.07*	90.85 ± 0.07

Method	Instance @5			Macro @5					
	P	PS	R	P	R	BA	F_1	JS	Cov
Synthetic Wiki10-31K									
Top-5	**80.42** ± **0.07**	107.38 ± 0.33	* **36.82** ± **0.05**	11.58 ± 0.04	4.47 ± 0.02	52.23 ± 0.01	5.77 ± 0.02	3.94 ± 0.02	14.98 ± 0.04
PS-5	*65.29* ± *0.09*	**245.24** ± **0.35**	*30.97* ± *0.04*	58.14 ± 0.15	68.79 ± 0.08	84.39 ± 0.04	58.06 ± 0.11	46.82 ± 0.11	89.24 ± 0.11
Pow-5$_{\beta=0.25}$	64.33 ± 0.09	*244.46* ± *0.42*	30.22 ± 0.04	57.97 ± 0.13	67.85 ± 0.10	83.92 ± 0.05	57.82 ± 0.11	46.63 ± 0.10	88.63 ± 0.11
Pow-5$_{\beta=0.5}$	57.78 ± 0.08	240.67 ± 0.36	26.07 ± 0.04	55.13 ± 0.13	69.86 ± 0.08	84.93 ± 0.04	57.01 ± 0.11	45.55 ± 0.10	89.73 ± 0.10
Log-5	62.97 ± 0.08	237.25 ± 0.35	28.16 ± 0.03	57.02 ± 0.13	63.46 ± 0.10	81.73 ± 0.05	55.63 ± 0.09	44.77 ± 0.08	85.45 ± 0.12
Macro-R$_{\text{prior}}$-5	48.22 ± 0.07	228.66 ± 0.45	19.74 ± 0.03	49.34 ± 0.15	70.47 ± 0.12	85.23 ± 0.06	53.21 ± 0.13	41.54 ± 0.11	89.97 ± 0.13
Macro-BA$_{\text{prior}}$-5	48.34 ± 0.08	228.80 ± 0.45	19.84 ± 0.03	49.38 ± 0.14	70.47 ± 0.11	85.23 ± 0.06	53.22 ± 0.13	41.55 ± 0.11	89.98 ± 0.13
BCA(Macro-P@5)	54.89 ± 0.06	136.56 ± 0.55	24.32 ± 0.02	**65.09** ± **0.22**	41.01 ± 0.09	70.50 ± 0.05	42.85 ± 0.14	32.79 ± 0.14	84.99 ± 0.21
BCA(Macro-R@5)	46.84 ± 0.09	223.63 ± 0.50	18.89 ± 0.04	48.61 ± 0.14	**72.24** ± **0.13**	**86.11** ± **0.07**	53.43 ± 0.13	41.48 ± 0.11	91.65 ± 0.14
BCA(Macro-BA@5)	46.91 ± 0.09	223.72 ± 0.50	18.94 ± 0.04	48.63 ± 0.14	**72.24** ± **0.13**	**86.11** ± **0.07**	53.43 ± 0.14	41.48 ± 0.12	91.66 ± 0.14
BCA(Macro-F_1@5)	58.45 ± 0.08	234.90 ± 0.42	26.30 ± 0.02	59.24 ± 0.17	69.65 ± 0.14	84.82 ± 0.07	**60.44** ± **0.15**	*48.54* ± *0.14*	*92.10* ± *0.14*
BCA(Macro-JS@5)	58.03 ± 0.08	235.73 ± 0.41	26.04 ± 0.02	58.81 ± 0.16	70.17 ± 0.12	85.08 ± 0.06	*60.31* ± *0.15*	**48.58** ± **0.14**	92.00 ± 0.14
BCA(Cov@5)	39.75 ± 0.09	200.86 ± 0.52	15.32 ± 0.03	45.30 ± 0.13	69.04 ± 0.14	84.52 ± 0.07	49.05 ± 0.13	37.02 ± 0.11	**92.59** ± **0.13**
FW(Macro-P@5)	55.28 ± 0.05	137.62 ± 0.37	24.48 ± 0.02	*62.12* ± *0.14*	40.24 ± 0.10	70.12 ± 0.05	41.41 ± 0.13	31.77 ± 0.14	81.43 ± 0.16
FW(Macro-R@5)	46.87 ± 0.10	225.21 ± 0.53	18.90 ± 0.04	48.68 ± 0.16	*71.05* ± *0.14*	*85.52* ± *0.07*	53.02 ± 0.14	41.23 ± 0.12	90.79 ± 0.15
FW(Macro-BA@5)	46.95 ± 0.10	225.30 ± 0.53	18.96 ± 0.04	48.71 ± 0.16	*71.05* ± *0.14*	*85.52* ± *0.07*	53.03 ± 0.14	41.24 ± 0.12	90.80 ± 0.15
FW(Macro-F_1@5)	58.47 ± 0.08	236.58 ± 0.44	26.30 ± 0.02	58.99 ± 0.20	68.19 ± 0.15	84.09 ± 0.07	59.18 ± 0.16	47.50 ± 0.15	90.87 ± 0.16
FW(Macro-JS@5)	58.10 ± 0.08	237.31 ± 0.43	26.07 ± 0.02	58.65 ± 0.20	68.96 ± 0.14	84.48 ± 0.07	59.30 ± 0.15	47.75 ± 0.14	90.85 ± 0.16
Synthetic WikiLSHTC-325K									
Top-5	**30.62** ± **0.01**	66.19 ± 0.02	* **66.39** ± **0.01**	29.00 ± 0.04	60.89 ± 0.05	80.44 ± 0.03	36.46 ± 0.05	25.57 ± 0.04	79.09 ± 0.07
PS-5	*30.07* ± *0.01*	**67.97** ± **0.02**	*65.75* ± *0.01*	27.15 ± 0.04	69.14 ± 0.06	84.57 ± 0.03	36.46 ± 0.05	25.33 ± 0.04	83.92 ± 0.08
Pow-5$_{\beta=0.25}$	29.36 ± 0.01	*67.60* ± *0.02*	64.69 ± 0.01	27.56 ± 0.04	68.90 ± 0.06	84.45 ± 0.03	36.93 ± 0.05	25.75 ± 0.04	83.68 ± 0.07
Pow-5$_{\beta=0.5}$	27.18 ± 0.01	65.76 ± 0.02	60.37 ± 0.01	23.85 ± 0.04	70.52 ± 0.07	85.26 ± 0.03	33.19 ± 0.05	22.61 ± 0.03	84.19 ± 0.08
Log-5	29.84 ± 0.01	67.36 ± 0.01	65.38 ± 0.01	28.97 ± 0.05	65.83 ± 0.06	82.91 ± 0.03	37.72 ± 0.05	26.50 ± 0.04	82.10 ± 0.08
Macro-R$_{\text{prior}}$-5	24.42 ± 0.01	62.44 ± 0.02	55.21 ± 0.01	19.39 ± 0.03	71.55 ± 0.07	85.77 ± 0.03	27.81 ± 0.03	18.40 ± 0.03	84.44 ± 0.08
Macro-BA$_{\text{prior}}$-5	24.42 ± 0.01	62.44 ± 0.02	55.21 ± 0.01	19.39 ± 0.03	71.55 ± 0.07	85.77 ± 0.03	27.81 ± 0.03	18.40 ± 0.03	84.44 ± 0.08
BCA(Macro-P@5)	12.48 ± 0.00	22.64 ± 0.02	25.90 ± 0.01	**50.21** ± **0.06**	29.82 ± 0.06	64.91 ± 0.03	27.75 ± 0.06	19.73 ± 0.05	75.65 ± 0.08
BCA(Macro-R@5)	22.11 ± 0.00	58.75 ± 0.02	50.63 ± 0.01	17.37 ± 0.01	**72.14** ± **0.05**	**86.07** ± **0.03**	24.61 ± 0.02	16.08 ± 0.01	*84.99* ± *0.07*
BCA(Macro-BA@5)	22.11 ± 0.00	58.75 ± 0.02	50.63 ± 0.01	17.37 ± 0.01	**72.14** ± **0.05**	**86.07** ± **0.03**	24.61 ± 0.02	16.08 ± 0.01	*84.99* ± *0.07*
BCA(Macro-F_1@5)	26.83 ± 0.01	60.44 ± 0.02	58.14 ± 0.01	40.93 ± 0.04	61.81 ± 0.04	80.90 ± 0.02	**47.11** ± **0.04**	**34.87** ± **0.03**	83.00 ± 0.07
BCA(Macro-JS@5)	26.80 ± 0.01	60.43 ± 0.01	58.04 ± 0.01	40.92 ± 0.04	61.81 ± 0.05	80.91 ± 0.02	*47.10* ± *0.04*	**34.87** ± **0.03**	82.99 ± 0.07
BCA(Cov@5)	5.40 ± 0.00	21.92 ± 0.01	11.72 ± 0.00	17.97 ± 0.01	56.52 ± 0.06	78.26 ± 0.03	16.81 ± 0.01	10.53 ± 0.01	**85.33** ± **0.06**
FW(Macro-P@5)	12.49 ± 0.00	22.68 ± 0.02	25.88 ± 0.00	*49.06* ± *0.06*	29.52 ± 0.06	64.76 ± 0.03	27.07 ± 0.06	19.28 ± 0.05	73.41 ± 0.07
FW(Macro-R@5)	22.11 ± 0.00	58.79 ± 0.02	50.63 ± 0.01	17.14 ± 0.01	*71.95* ± *0.06*	*85.97* ± *0.03*	24.23 ± 0.02	15.84 ± 0.01	84.83 ± 0.08
FW(Macro-BA@5)	22.11 ± 0.00	58.79 ± 0.02	50.63 ± 0.01	17.14 ± 0.01	*71.95* ± *0.05*	*85.97* ± *0.03*	24.23 ± 0.02	15.84 ± 0.01	84.83 ± 0.08
FW(Macro-F_1@5)	26.83 ± 0.01	60.42 ± 0.01	58.13 ± 0.01	40.88 ± 0.04	61.40 ± 0.05	80.70 ± 0.02	46.84 ± 0.04	*34.68* ± *0.03*	82.65 ± 0.07
FW(Macro-JS@5)	26.80 ± 0.01	60.39 ± 0.01	58.02 ± 0.01	40.88 ± 0.04	61.40 ± 0.05	80.70 ± 0.02	46.83 ± 0.04	*34.68* ± *0.03*	82.64 ± 0.07

Method	Instance @5			Macro @5					
	P	PS	R	P	R	BA	F_1	JS	Cov
Synthetic WikipediaLarge-500K									
Top-5	**40.85** ± 0.01	94.06 ± 0.03	* **66.37** ± 0.01	36.15 ± 0.02	57.46 ± 0.01	78.73 ± 0.00	40.26 ± 0.02	28.53 ± 0.02	82.97 ± 0.02
PS-5	39.22 ± 0.01	**101.28** ± 0.03	65.04 ± 0.01	37.59 ± 0.02	75.74 ± 0.01	87.87 ± 0.00	47.19 ± 0.02	33.84 ± 0.02	93.84 ± 0.01
Pow-5$_{\beta=0.25}$	38.71 ± 0.01	*100.97* ± *0.02*	64.39 ± 0.01	38.15 ± 0.02	75.03 ± 0.01	87.52 ± 0.00	47.66 ± 0.02	34.30 ± 0.02	93.52 ± 0.01
Pow-5$_{\beta=0.5}$	36.73 ± 0.01	99.51 ± 0.02	61.36 ± 0.01	34.32 ± 0.02	77.26 ± 0.01	88.63 ± 0.00	44.54 ± 0.02	31.38 ± 0.02	94.08 ± 0.01
Log-5	*39.65* ± *0.01*	99.69 ± 0.03	*65.28* ± *0.01*	39.33 ± 0.02	68.74 ± 0.01	84.37 ± 0.01	46.76 ± 0.02	33.74 ± 0.03	90.67 ± 0.01
Macro-R$_{\text{prior}}$-5	33.09 ± 0.01	95.08 ± 0.03	55.67 ± 0.01	29.36 ± 0.02	78.33 ± 0.01	89.16 ± 0.00	39.39 ± 0.02	26.86 ± 0.02	94.30 ± 0.00
Macro-BA$_{\text{prior}}$-5	33.10 ± 0.01	95.09 ± 0.03	55.68 ± 0.01	29.36 ± 0.02	78.33 ± 0.01	89.16 ± 0.00	39.39 ± 0.02	26.86 ± 0.02	94.30 ± 0.00
BCA(Macro-P@5)	18.09 ± 0.00	35.29 ± 0.01	26.54 ± 0.00	**62.69** ± 0.03	29.61 ± 0.01	64.80 ± 0.01	31.45 ± 0.02	22.01 ± 0.02	87.59 ± 0.02
BCA(Macro-R@5)	30.78 ± 0.01	91.07 ± 0.03	51.96 ± 0.01	26.83 ± 0.02	**79.06** ± 0.01	**89.53** ± 0.00	35.80 ± 0.02	24.03 ± 0.02	*94.80* ± *0.00*
BCA(Macro-BA@5)	30.78 ± 0.01	91.08 ± 0.03	51.96 ± 0.01	26.83 ± 0.02	**79.06** ± 0.01	**89.53** ± 0.00	35.80 ± 0.02	24.03 ± 0.02	*94.80* ± *0.00*
BCA(Macro-F_1@5)	36.12 ± 0.01	93.43 ± 0.02	58.71 ± 0.01	52.39 ± 0.03	70.51 ± 0.01	85.25 ± 0.00	**57.81** ± 0.02	*44.01* ± *0.03*	93.47 ± 0.01
BCA(Macro-JS@5)	36.08 ± 0.01	93.43 ± 0.02	58.59 ± 0.01	52.34 ± 0.03	70.53 ± 0.01	85.26 ± 0.00	*57.79* ± *0.02*	**44.02** ± 0.02	93.45 ± 0.01
BCA(Cov@5)	10.03 ± 0.00	40.94 ± 0.02	15.85 ± 0.01	24.43 ± 0.02	61.36 ± 0.01	80.68 ± 0.00	23.60 ± 0.01	14.93 ± 0.01	**95.11** ± 0.00
FW(Macro-P@5)	18.08 ± 0.00	35.30 ± 0.01	26.45 ± 0.00	*61.20* ± *0.04*	29.12 ± 0.01	64.56 ± 0.01	30.47 ± 0.02	21.40 ± 0.02	84.53 ± 0.03
FW(Macro-R@5)	30.77 ± 0.01	91.13 ± 0.03	51.95 ± 0.01	26.69 ± 0.02	*78.85* ± *0.01*	*89.43* ± *0.00*	35.61 ± 0.02	23.86 ± 0.02	94.68 ± 0.00
FW(Macro-BA@5)	30.78 ± 0.01	91.13 ± 0.03	51.96 ± 0.01	26.69 ± 0.02	*78.85* ± *0.01*	*89.43* ± *0.00*	35.61 ± 0.02	23.86 ± 0.02	94.68 ± 0.00
FW(Macro-F_1@5)	36.11 ± 0.01	93.40 ± 0.02	58.69 ± 0.01	52.39 ± 0.03	70.13 ± 0.01	85.06 ± 0.01	57.59 ± 0.03	43.84 ± 0.03	93.19 ± 0.02
FW(Macro-JS@5)	36.08 ± 0.01	93.41 ± 0.02	58.58 ± 0.01	52.34 ± 0.03	70.15 ± 0.01	85.07 ± 0.01	57.56 ± 0.03	43.85 ± 0.03	93.17 ± 0.02
Synthetic Amazon-670K									
Top-5	**42.57** ± 0.01	246.20 ± 0.08	* **62.77** ± 0.01	41.75 ± 0.02	67.00 ± 0.02	83.50 ± 0.01	48.10 ± 0.02	35.83 ± 0.02	86.01 ± 0.01
PS-5	41.55 ± 0.01	**256.18** ± 0.08	61.32 ± 0.01	41.29 ± 0.02	73.46 ± 0.03	86.73 ± 0.01	49.51 ± 0.02	36.60 ± 0.02	91.20 ± 0.01
Pow-5$_{\beta=0.25}$	41.99 ± 0.01	255.56 ± 0.09	61.96 ± 0.01	41.99 ± 0.02	72.39 ± 0.03	86.20 ± 0.01	49.87 ± 0.03	37.03 ± 0.02	90.49 ± 0.01
Pow-5$_{\beta=0.5}$	41.25 ± 0.01	*255.95* ± *0.08*	60.86 ± 0.00	40.87 ± 0.02	73.75 ± 0.02	86.87 ± 0.01	49.15 ± 0.02	36.22 ± 0.02	91.39 ± 0.02
Log-5	*42.44* ± *0.01*	251.91 ± 0.08	*62.59* ± *0.01*	42.22 ± 0.02	69.76 ± 0.03	84.88 ± 0.01	49.31 ± 0.02	36.74 ± 0.02	88.41 ± 0.02
Macro-R$_{\text{prior}}$-5	39.97 ± 0.01	253.53 ± 0.06	58.83 ± 0.01	39.42 ± 0.02	74.27 ± 0.02	87.13 ± 0.01	47.61 ± 0.02	34.75 ± 0.02	91.66 ± 0.02
Macro-BA$_{\text{prior}}$-5	39.97 ± 0.01	253.53 ± 0.06	58.83 ± 0.01	39.42 ± 0.02	74.27 ± 0.02	87.13 ± 0.01	47.61 ± 0.02	34.75 ± 0.02	91.66 ± 0.02
BCA(Macro-P@5)	21.46 ± 0.01	124.58 ± 0.02	29.21 ± 0.02	**56.48** ± 0.02	39.63 ± 0.02	69.82 ± 0.01	40.09 ± 0.01	29.22 ± 0.01	82.85 ± 0.02
BCA(Macro-R@5)	39.23 ± 0.01	249.80 ± 0.07	57.68 ± 0.01	38.64 ± 0.02	**75.05** ± 0.02	**87.52** ± 0.01	46.83 ± 0.02	33.98 ± 0.02	*92.45* ± *0.02*
BCA(Macro-BA@5)	39.23 ± 0.01	249.80 ± 0.07	57.68 ± 0.01	38.64 ± 0.02	**75.05** ± 0.02	**87.52** ± 0.01	46.83 ± 0.02	33.98 ± 0.02	*92.45* ± *0.02*
BCA(Macro-F_1@5)	39.91 ± 0.01	242.67 ± 0.07	57.03 ± 0.02	50.02 ± 0.02	70.14 ± 0.01	85.07 ± 0.01	**55.56** ± 0.02	*42.38* ± *0.02*	91.91 ± 0.02
BCA(Macro-JS@5)	39.92 ± 0.01	243.11 ± 0.07	57.02 ± 0.02	49.86 ± 0.02	70.20 ± 0.01	85.10 ± 0.01	*55.48* ± *0.02*	**42.39** ± 0.02	91.76 ± 0.01
BCA(Cov@5)	26.58 ± 0.01	195.18 ± 0.11	41.06 ± 0.02	39.77 ± 0.02	67.92 ± 0.03	83.96 ± 0.01	42.03 ± 0.02	29.75 ± 0.02	**93.59** ± 0.02
FW(Macro-P@5)	21.17 ± 0.01	124.47 ± 0.04	28.85 ± 0.01	*54.67* ± *0.02*	39.03 ± 0.02	69.51 ± 0.01	38.84 ± 0.01	28.35 ± 0.01	80.31 ± 0.03
FW(Macro-R@5)	39.25 ± 0.01	250.58 ± 0.06	57.70 ± 0.01	38.98 ± 0.02	*74.64* ± *0.01*	*87.32* ± *0.01*	46.91 ± 0.02	34.10 ± 0.02	92.09 ± 0.02
FW(Macro-BA@5)	39.25 ± 0.01	250.58 ± 0.06	57.70 ± 0.01	38.98 ± 0.02	*74.64* ± *0.01*	*87.32* ± *0.01*	46.91 ± 0.02	34.10 ± 0.02	92.09 ± 0.02
FW(Macro-F_1@5)	39.82 ± 0.01	242.70 ± 0.08	56.83 ± 0.01	49.96 ± 0.03	69.33 ± 0.02	84.66 ± 0.01	54.96 ± 0.02	41.95 ± 0.02	91.08 ± 0.02
FW(Macro-JS@5)	39.83 ± 0.01	243.04 ± 0.07	56.82 ± 0.01	49.83 ± 0.02	69.41 ± 0.02	84.71 ± 0.01	54.93 ± 0.02	42.00 ± 0.02	90.91 ± 0.01

B.2.2 Comparison of inference algorithms on original datasets

Table B.3: Results (%) for $k \in \{1, 3, 5\}$ on original XMLC datasets with marginal conditional probabilities coming from PLT model. Each experiments was repeated 5 times with mean and standard deviation reported after the ± sign. The green background indicates cells in which the inference algorithm matches the metric it optimizes and the gray text indicates results worse than those results for a given metric. The best results are in **bold**, and the second best are in *italic*. * – while Top-k in general is not optimal for recall@k, we expect it to be the closest to the optimal solution, and we mark it blue background.

Method	Instance @1			Macro @1					
	P	PS	R	P	R	BA	F_1	JS	Cov
RCV1x-2K									
Top-1	**89.99**	*96.94*	* **40.31**	10.04	1.24	50.62	1.74	1.18	12.42
PS-1	*88.16*	**97.38**	*39.60*	24.13	3.28	51.64	5.05	3.10	36.24
Pow-$1_{\beta=0.25}$	83.82	94.05	37.72	23.00	4.06	52.02	6.17	3.74	40.55
Pow-$1_{\beta=0.5}$	75.76	86.98	34.04	22.35	6.23	53.11	8.55	5.14	50.46
Log-1	84.23	94.16	37.87	21.52	3.43	51.71	5.25	3.22	35.59
Macro-R_{prior}-1	49.09	59.67	21.12	16.37	9.66	54.82	8.77	5.15	61.16
Macro-BA_{prior}-1	49.89	60.52	21.55	16.45	9.66	54.82	8.80	5.17	61.16
BCA(Macro-P@1)	62.22 ± 0.42	66.17 ± 0.44	24.85 ± 0.35	**34.89 ± 0.13**	2.05 ± 0.06	51.02 ± 0.03	2.95 ± 0.03	1.81 ± 0.03	36.24 ± 0.08
BCA(Macro-R@1)	20.16 ± 0.00	27.82 ± 0.00	6.64 ± 0.00	14.81 ± 0.00	**9.98 ± 0.00**	*54.97 ± 0.00*	7.02 ± 0.00	4.00 ± 0.00	*64.58 ± 0.00*
BCA(Macro-BA@1)	39.18 ± 0.00	48.74 ± 0.00	16.13 ± 0.00	15.26 ± 0.00	**9.98 ± 0.00**	**54.98 ± 0.00**	8.06 ± 0.00	4.70 ± 0.00	63.64 ± 0.00
BCA(Macro-F_1@1)	70.26 ± 0.01	80.95 ± 0.01	31.35 ± 0.00	23.68 ± 0.08	6.65 ± 0.04	53.32 ± 0.02	**9.70 ± 0.04**	*5.71 ± 0.03*	57.30 ± 0.20
BCA(Macro-JS@1)	70.94 ± 0.00	81.75 ± 0.01	31.84 ± 0.00	23.45 ± 0.08	6.59 ± 0.04	53.29 ± 0.02	*9.59 ± 0.05*	**5.75 ± 0.03**	56.16 ± 0.13
BCA(Cov@1)	3.20 ± 0.01	7.27 ± 0.02	0.41 ± 0.01	18.50 ± 0.07	7.83 ± 0.03	53.90 ± 0.02	4.66 ± 0.01	2.54 ± 0.01	**69.41 ± 0.20**
FW(Macro-P@1)	59.90 ± 0.02	67.49 ± 0.03	24.52 ± 0.01	*25.20 ± 0.05*	3.80 ± 0.01	51.89 ± 0.01	5.94 ± 0.02	3.41 ± 0.01	43.93 ± 0.09
FW(Macro-R@1)	48.75 ± 0.00	59.32 ± 0.00	20.95 ± 0.00	16.38 ± 0.00	*9.72 ± 0.00*	54.85 ± 0.00	8.78 ± 0.00	5.14 ± 0.00	61.32 ± 0.00
FW(Macro-BA@1)	49.58 ± 0.00	60.20 ± 0.00	21.40 ± 0.00	16.43 ± 0.00	*9.72 ± 0.00*	54.85 ± 0.00	8.81 ± 0.00	5.17 ± 0.00	61.32 ± 0.00
FW(Macro-F_1@1)	71.91 ± 0.07	82.44 ± 0.07	31.82 ± 0.04	22.13 ± 0.07	5.41 ± 0.01	52.70 ± 0.00	8.03 ± 0.01	4.71 ± 0.01	48.91 ± 0.05
FW(Macro-JS@1)	72.82 ± 0.00	83.51 ± 0.00	32.54 ± 0.00	21.81 ± 0.04	5.46 ± 0.00	52.72 ± 0.00	8.05 ± 0.00	4.76 ± 0.00	48.39 ± 0.06

Method	Instance @1 P	Instance @1 PS	Instance @1 R	Macro @1 P	Macro @1 R	Macro @1 BA	Macro @1 F_1	Macro @1 JS	Macro @1 Cov
EURLex-4K									
Top-1	**81.65**	180.02	* **16.45**	14.29	4.83	52.41	6.61	4.68	16.30
PS-1	78.34	*224.76*	15.77	22.39	9.98	54.99	12.42	9.44	25.52
Pow-1$_{\beta=0.25}$	79.15	218.13	15.92	21.04	9.04	54.52	11.45	8.63	24.02
Pow-1$_{\beta=0.5}$	74.56	**231.73**	15.02	23.77	11.44	55.71	13.81	10.68	27.55
Log-1	*79.65*	210.24	*16.03*	19.79	8.09	54.04	10.45	7.76	22.59
Macro-R$_{\text{prior}}$-1	57.05	214.35	11.42	23.19	12.86	56.43	14.42	11.23	27.97
Macro-BA$_{\text{prior}}$-1	57.18	214.60	11.44	23.19	12.86	56.43	14.42	11.23	27.97
BCA(Macro-P@1)	43.17 ± 0.46	153.63 ± 1.45	8.56 ± 0.10	**31.87 ± 0.16**	9.80 ± 0.15	54.89 ± 0.08	12.93 ± 0.15	9.70 ± 0.15	32.42 ± 0.18
BCA(Macro-R@1)	57.36 ± 0.00	215.49 ± 0.00	11.48 ± 0.00	24.68 ± 0.00	**13.48 ± 0.00**	**56.73 ± 0.00**	15.34 ± 0.00	**11.97 ± 0.00**	29.70 ± 0.00
BCA(Macro-BA@1)	57.42 ± 0.00	215.60 ± 0.00	11.49 ± 0.00	24.68 ± 0.00	**13.48 ± 0.00**	**56.73 ± 0.00**	15.34 ± 0.00	**11.97 ± 0.00**	29.70 ± 0.00
BCA(Macro-F_1@1)	62.20 ± 0.15	208.89 ± 0.57	12.50 ± 0.02	*29.16 ± 0.12*	11.92 ± 0.06	55.95 ± 0.03	**15.37 ± 0.07**	11.65 ± 0.06	*32.90 ± 0.15*
BCA(Macro-JS@1)	65.12 ± 0.19	217.02 ± 0.75	13.13 ± 0.04	28.05 ± 0.17	12.10 ± 0.06	56.04 ± 0.03	*15.36 ± 0.07*	*11.78 ± 0.06*	31.50 ± 0.17
BCA(Cov@1)	50.49 ± 0.25	181.84 ± 1.01	9.96 ± 0.07	27.55 ± 0.15	11.07 ± 0.10	55.53 ± 0.05	13.61 ± 0.09	10.01 ± 0.08	**34.21 ± 0.22**
FW(Macro-P@1)	75.85 ± 0.18	205.61 ± 0.59	15.24 ± 0.03	**21.84 ± 0.08**	8.54 ± 0.02	54.27 ± 0.01	11.14 ± 0.03	8.20 ± 0.02	24.81 ± 0.07
FW(Macro-R@1)	57.92 ± 0.00	216.62 ± 0.00	11.59 ± 0.00	23.42 ± 0.00	*12.94 ± 0.00*	*56.46 ± 0.00*	14.52 ± 0.00	11.30 ± 0.00	28.22 ± 0.00
FW(Macro-BA@1)	58.02 ± 0.00	216.76 ± 0.00	11.61 ± 0.00	23.43 ± 0.00	12.93 ± 0.00	*56.46 ± 0.00*	14.53 ± 0.00	11.31 ± 0.00	28.22 ± 0.00
FW(Macro-F_1@1)	63.31 ± 0.13	213.07 ± 0.32	12.61 ± 0.03	23.53 ± 0.19	11.43 ± 0.02	55.71 ± 0.01	**13.61 ± 0.04**	10.36 ± 0.02	28.63 ± 0.22
FW(Macro-JS@1)	66.09 ± 0.01	223.13 ± 0.03	13.28 ± 0.00	23.85 ± 0.00	12.02 ± 0.00	56.01 ± 0.00	14.16 ± 0.00	**10.95 ± 0.00**	28.40 ± 0.00
EURLex-4.3K									
Top-1	**91.40**	151.59	* **20.81**	14.39	4.45	52.22	6.16	4.37	15.62
PS-1	88.13	*188.97*	20.02	24.11	10.84	55.42	13.46	10.40	26.55
Pow-1$_{\beta=0.25}$	88.40	186.24	20.03	23.03	10.12	55.06	12.71	9.75	25.47
Pow-1$_{\beta=0.5}$	83.95	**191.93**	19.04	24.66	12.22	56.11	14.68	11.50	28.05
Log-1	*89.45*	181.73	*20.32*	21.53	9.00	54.50	11.51	8.72	23.65
Macro-R$_{\text{prior}}$-1	70.18	179.57	15.75	24.71	13.56	56.78	15.40	12.06	29.50
Macro-BA$_{\text{prior}}$-1	70.30	179.79	15.80	24.75	13.58	56.79	15.43	12.09	29.52
BCA(Macro-P@1)	59.90 ± 0.59	131.49 ± 0.91	13.25 ± 0.15	**31.71 ± 0.12**	9.73 ± 0.06	54.86 ± 0.03	12.93 ± 0.08	9.61 ± 0.06	32.47 ± 0.12
BCA(Macro-R@1)	61.82 ± 0.00	164.09 ± 0.00	13.87 ± 0.00	24.92 ± 0.00	**13.93 ± 0.00**	**56.96 ± 0.00**	15.65 ± 0.00	12.26 ± 0.00	29.95 ± 0.00
BCA(Macro-BA@1)	61.87 ± 0.00	164.15 ± 0.00	13.88 ± 0.00	24.96 ± 0.00	**13.93 ± 0.00**	**56.96 ± 0.00**	15.66 ± 0.00	12.26 ± 0.00	29.97 ± 0.00
BCA(Macro-F_1@1)	76.55 ± 0.07	178.92 ± 0.06	17.31 ± 0.02	*29.54 ± 0.07*	12.72 ± 0.01	56.36 ± 0.01	**16.24 ± 0.01**	*12.36 ± 0.01*	*33.11 ± 0.08*
BCA(Macro-JS@1)	77.70 ± 0.05	182.29 ± 0.24	17.58 ± 0.01	28.04 ± 0.06	12.83 ± 0.04	56.41 ± 0.02	*16.01 ± 0.05*	**12.39 ± 0.04**	31.62 ± 0.06
BCA(Cov@1)	42.05 ± 0.20	122.25 ± 0.31	8.89 ± 0.06	27.22 ± 0.13	11.65 ± 0.05	55.82 ± 0.03	13.40 ± 0.06	9.82 ± 0.05	**34.68 ± 0.12**
FW(Macro-P@1)	85.85 ± 0.09	176.35 ± 0.33	19.55 ± 0.02	**23.80 ± 0.08**	9.43 ± 0.04	54.71 ± 0.02	12.28 ± 0.04	9.15 ± 0.04	26.03 ± 0.09
FW(Macro-R@1)	71.12 ± 0.00	181.45 ± 0.00	15.99 ± 0.00	24.69 ± 0.00	13.65 ± 0.00	*56.82 ± 0.00*	15.49 ± 0.00	12.16 ± 0.00	29.55 ± 0.00
FW(Macro-BA@1)	71.22 ± 0.00	181.59 ± 0.00	16.03 ± 0.00	24.71 ± 0.00	*13.66 ± 0.00*	*56.82 ± 0.00*	15.50 ± 0.00	12.17 ± 0.00	29.55 ± 0.00
FW(Macro-F_1@1)	77.53 ± 0.24	183.57 ± 1.11	17.55 ± 0.06	25.80 ± 0.21	12.62 ± 0.17	56.31 ± 0.08	**14.99 ± 0.14**	11.62 ± 0.12	29.99 ± 0.22
FW(Macro-JS@1)	74.31 ± 0.01	183.42 ± 0.04	16.75 ± 0.00	25.02 ± 0.01	13.21 ± 0.00	56.60 ± 0.00	15.16 ± 0.00	**11.83 ± 0.00**	29.56 ± 0.01

Method	Instance @1			Macro @1					
	P	PS	R	P	R	BA	F_1	JS	Cov
AmazonCat-13K									
Top-1	**93.09**	101.55	* **26.37**	15.40	1.68	50.84	2.75	1.65	17.54
PS-1	*87.66*	**135.18**	*24.57*	56.29	28.40	64.20	33.92	25.96	69.82
Pow-$1_{\beta=0.25}$	79.30	*129.40*	21.45	53.20	27.54	63.77	33.15	25.05	68.03
Pow-$1_{\beta=0.5}$	70.07	128.12	18.27	52.09	35.40	67.70	38.52	29.71	74.88
Log-1	78.91	121.35	21.00	48.41	19.93	59.97	25.85	18.75	59.07
Macro-R_{prior}-1	55.40	117.65	13.41	45.21	42.35	71.17	38.81	29.57	80.02
Macro-BA_{prior}-1	55.63	117.89	13.51	45.23	42.35	71.17	38.82	29.59	80.02
BCA(Macro-P@1)	75.27 ± 0.05	89.75 ± 0.05	20.92 ± 0.01	**70.53 ± 0.03**	13.45 ± 0.01	56.73 ± 0.00	18.53 ± 0.01	13.41 ± 0.01	71.05 ± 0.03
BCA(Macro-R@1)	48.21 ± 0.00	109.24 ± 0.00	11.23 ± 0.00	43.19 ± 0.00	**44.15 ± 0.00**	**72.07 ± 0.00**	37.51 ± 0.00	28.14 ± 0.00	82.54 ± 0.00
BCA(Macro-BA@1)	48.42 ± 0.00	109.46 ± 0.00	11.32 ± 0.00	43.22 ± 0.00	**44.15 ± 0.00**	**72.07 ± 0.00**	37.52 ± 0.00	28.15 ± 0.00	*82.56 ± 0.00*
BCA(Macro-F_1@1)	69.49 ± 0.00	123.53 ± 0.01	18.10 ± 0.00	*59.75 ± 0.02*	33.62 ± 0.02	66.81 ± 0.01	**40.43 ± 0.01**	*30.47 ± 0.02*	79.87 ± 0.02
BCA(Macro-JS@1)	69.06 ± 0.00	123.79 ± 0.01	17.94 ± 0.00	58.67 ± 0.02	33.74 ± 0.01	66.87 ± 0.01	*40.21 ± 0.01*	**30.57 ± 0.01**	79.17 ± 0.04
BCA(Cov@1)	14.68 ± 0.02	51.60 ± 0.04	2.65 ± 0.00	41.13 ± 0.04	33.95 ± 0.02	66.97 ± 0.01	25.12 ± 0.02	17.23 ± 0.02	**85.41 ± 0.04**
FW(Macro-P@1)	77.09 ± 0.02	117.41 ± 0.03	21.13 ± 0.01	58.54 ± 0.01	24.36 ± 0.02	62.18 ± 0.01	31.22 ± 0.01	22.89 ± 0.02	69.99 ± 0.02
FW(Macro-R@1)	55.02 ± 0.00	117.36 ± 0.00	13.31 ± 0.00	44.91 ± 0.00	*42.81 ± 0.00*	*71.40 ± 0.00*	38.72 ± 0.00	29.42 ± 0.00	80.47 ± 0.00
FW(Macro-BA@1)	55.26 ± 0.00	117.60 ± 0.00	13.41 ± 0.00	44.93 ± 0.00	*42.81 ± 0.00*	*71.40 ± 0.00*	38.73 ± 0.00	29.43 ± 0.00	80.48 ± 0.00
FW(Macro-F_1@1)	55.76 ± 0.01	115.87 ± 0.03	13.37 ± 0.01	46.38 ± 0.07	40.26 ± 0.03	70.13 ± 0.01	37.93 ± 0.04	28.56 ± 0.04	80.40 ± 0.11
FW(Macro-JS@1)	56.30 ± 0.00	117.77 ± 0.00	13.76 ± 0.00	45.99 ± 0.00	41.37 ± 0.00	70.69 ± 0.00	38.45 ± 0.00	29.30 ± 0.00	79.42 ± 0.00
AmazonCat-14K									
Top-1	**89.28**	94.08	* **39.40**	25.19	2.74	51.37	4.54	2.66	30.25
PS-1	*85.96*	**107.81**	*38.35*	52.00	25.13	62.56	30.79	21.99	70.52
Pow-$1_{\beta=0.25}$	80.24	*105.84*	36.06	48.68	27.76	63.88	32.72	23.57	71.39
Pow-$1_{\beta=0.5}$	71.03	100.15	30.73	42.86	34.96	67.48	35.43	**25.84**	76.89
Log-1	81.90	103.84	36.42	48.07	20.61	60.31	26.75	18.60	64.52
Macro-R_{prior}-1	46.81	77.05	13.25	33.65	40.19	70.10	31.79	22.63	80.63
Macro-BA_{prior}-1	47.45	77.71	13.86	33.66	40.19	70.10	31.80	22.64	80.63
BCA(Macro-P@1)	63.05 ± 0.00	66.73 ± 0.00	30.65 ± 0.00	**64.30 ± 0.02**	6.70 ± 0.01	53.35 ± 0.01	9.94 ± 0.01	6.66 ± 0.01	66.24 ± 0.01
BCA(Macro-R@1)	42.58 ± 0.02	71.95 ± 0.01	11.31 ± 0.02	32.11 ± 0.00	**41.33 ± 0.00**	**70.66 ± 0.00**	29.99 ± 0.00	21.13 ± 0.00	*82.25 ± 0.01*
BCA(Macro-BA@1)	42.68 ± 0.04	72.05 ± 0.04	11.40 ± 0.03	32.11 ± 0.01	*41.32 ± 0.00*	**70.66 ± 0.00**	29.99 ± 0.00	21.13 ± 0.00	82.24 ± 0.01
BCA(Macro-F_1@1)	70.35 ± 0.00	97.13 ± 0.00	29.56 ± 0.00	48.31 ± 0.01	31.28 ± 0.01	65.64 ± 0.00	**35.97 ± 0.00**	25.59 ± 0.00	78.47 ± 0.03
BCA(Macro-JS@1)	71.01 ± 0.00	98.18 ± 0.01	30.51 ± 0.00	47.75 ± 0.04	31.35 ± 0.01	65.68 ± 0.01	*35.90 ± 0.02*	*25.75 ± 0.01*	77.99 ± 0.03
BCA(Cov@1)	6.89 ± 0.00	18.47 ± 0.00	1.53 ± 0.00	29.68 ± 0.05	26.29 ± 0.00	63.14 ± 0.00	13.85 ± 0.00	8.42 ± 0.00	**83.81 ± 0.00**
FW(Macro-P@1)	66.03 ± 0.01	80.86 ± 0.01	30.54 ± 0.00	*55.41 ± 0.03*	18.20 ± 0.01	59.10 ± 0.01	24.97 ± 0.01	16.74 ± 0.00	70.71 ± 0.01
FW(Macro-R@1)	45.55 ± 0.00	75.78 ± 0.00	12.30 ± 0.00	33.18 ± 0.00	40.70 ± 0.00	*70.35 ± 0.00*	31.44 ± 0.00	22.30 ± 0.00	81.11 ± 0.00
FW(Macro-BA@1)	46.06 ± 0.00	76.29 ± 0.00	12.78 ± 0.00	33.19 ± 0.00	40.70 ± 0.00	*70.35 ± 0.00*	31.45 ± 0.00	22.31 ± 0.00	81.11 ± 0.00
FW(Macro-F_1@1)	69.84 ± 0.06	95.79 ± 0.13	28.38 ± 0.04	48.71 ± 0.01	28.96 ± 0.00	64.48 ± 0.00	34.21 ± 0.00	24.34 ± 0.00	74.47 ± 0.02
FW(Macro-JS@1)	53.70 ± 0.00	83.38 ± 0.00	17.61 ± 0.00	38.80 ± 0.00	35.59 ± 0.00	67.79 ± 0.00	33.49 ± 0.00	24.30 ± 0.00	76.54 ± 0.00

Method	Instance @1 P	Instance @1 PS	Instance @1 R	Macro @1 P	Macro @1 R	Macro @1 BA	Macro @1 F_1	Macro @1 JS	Macro @1 Cov
Wiki10-31K									
Top-1	**83.86**	108.68	* **4.93**	1.08	0.09	50.04	0.16	0.09	1.13
PS-1	*80.67*	194.18	4.82	4.61	1.36	50.68	1.79	1.34	4.97
Pow-1$_{\beta=0.25}$	80.49	192.23	*4.83*	4.35	1.18	50.59	1.61	1.16	4.75
Pow-1$_{\beta=0.5}$	72.84	223.37	4.35	5.60	1.97	50.98	2.50	1.92	6.17
Log-1	78.43	181.15	4.70	3.87	0.93	50.47	1.32	0.92	4.29
Macro-R$_{\text{prior}}$-1	53.63	*241.95*	3.14	6.70	**3.11**	**51.55**	**3.62**	**2.96**	7.45
Macro-BA$_{\text{prior}}$-1	53.84	**242.00**	3.15	6.70	**3.11**	**51.55**	**3.62**	**2.96**	7.45
BCA(Macro-P@1)	50.54 ± 0.29	180.21 ± 1.56	2.92 ± 0.01	**8.01 ± 0.08**	1.96 ± 0.03	50.98 ± 0.01	2.67 ± 0.03	1.96 ± 0.03	8.03 ± 0.08
BCA(Macro-R@1)	37.18 ± 0.00	195.80 ± 0.00	2.14 ± 0.00	6.35 ± 0.00	*2.82 ± 0.00*	*51.41 ± 0.00*	3.42 ± 0.00	*2.75 ± 0.00*	6.86 ± 0.00
BCA(Macro-BA@1)	37.23 ± 0.00	195.85 ± 0.00	2.14 ± 0.00	6.35 ± 0.00	*2.82 ± 0.00*	*51.41 ± 0.00*	3.42 ± 0.00	*2.75 ± 0.00*	6.85 ± 0.00
BCA(Macro-F_1@1)	50.49 ± 0.11	204.40 ± 0.84	2.95 ± 0.01	**8.01 ± 0.02**	2.32 ± 0.01	51.16 ± 0.01	3.10 ± 0.01	2.32 ± 0.01	**8.40 ± 0.01**
BCA(Macro-JS@1)	52.05 ± 0.08	208.60 ± 0.79	3.05 ± 0.00	*7.87 ± 0.02*	2.32 ± 0.01	51.16 ± 0.01	3.11 ± 0.01	2.31 ± 0.01	*8.34 ± 0.01*
BCA(Cov@1)	49.66 ± 0.30	180.54 ± 0.93	2.88 ± 0.02	6.67 ± 0.04	1.67 ± 0.01	50.83 ± 0.00	2.32 ± 0.01	1.64 ± 0.01	7.70 ± 0.03
FW(Macro-P@1)	56.43 ± 0.48	127.83 ± 1.76	3.28 ± 0.04	3.28 ± 0.07	1.08 ± 0.04	50.54 ± 0.02	1.32 ± 0.04	1.05 ± 0.04	3.55 ± 0.06
FW(Macro-R@1)	53.63 ± 0.00	*241.95 ± 0.00*	3.14 ± 0.00	6.70 ± 0.00	**3.11 ± 0.00**	**51.55 ± 0.00**	**3.62 ± 0.00**	**2.96 ± 0.00**	7.45 ± 0.00
FW(Macro-BA@1)	53.84 ± 0.00	**242.00 ± 0.00**	3.15 ± 0.00	6.70 ± 0.00	**3.11 ± 0.00**	**51.55 ± 0.00**	**3.62 ± 0.00**	**2.96 ± 0.00**	7.45 ± 0.00
FW(Macro-F_1@1)	57.68 ± 0.07	218.81 ± 1.04	3.40 ± 0.01	6.05 ± 0.02	2.25 ± 0.02	51.12 ± 0.01	2.82 ± 0.02	2.15 ± 0.02	6.98 ± 0.02
FW(Macro-JS@1)	52.26 ± 0.01	238.92 ± 0.01	3.05 ± 0.00	6.65 ± 0.00	**3.11 ± 0.00**	**51.55 ± 0.00**	*3.61 ± 0.00*	**2.96 ± 0.00**	7.41 ± 0.00
WikiLSHTC-325K									
Top-1	*63.44*	163.80	* 28.11	12.83	6.53	53.27	7.78	5.93	16.61
PS-1	**64.60**	207.38	**29.19**	19.12	10.58	55.29	12.39	9.63	24.47
Pow-1$_{\beta=0.25}$	62.41	200.99	*28.50*	17.92	9.84	54.92	11.57	8.93	23.24
Pow-1$_{\beta=0.5}$	61.53	221.59	28.49	20.77	12.31	56.16	14.14	11.06	27.37
Log-1	61.71	184.87	28.03	15.57	8.18	54.09	9.72	7.42	20.23
Macro-R$_{\text{prior}}$-1	57.87	240.94	27.31	23.40	15.47	57.74	16.95	13.36	32.31
Macro-BA$_{\text{prior}}$-1	57.87	240.94	27.31	23.40	15.47	57.74	16.95	13.36	32.31
BCA(Macro-P@1)	28.83 ± 0.04	124.49 ± 0.08	10.76 ± 0.03	**32.33 ± 0.01**	10.78 ± 0.01	55.39 ± 0.01	13.90 ± 0.01	10.62 ± 0.01	32.75 ± 0.02
BCA(Macro-R@1)	47.68 ± 0.00	235.46 ± 0.01	23.56 ± 0.00	26.38 ± 0.00	**18.19 ± 0.00**	**59.09 ± 0.00**	**19.08 ± 0.00**	**14.96 ± 0.00**	**36.75 ± 0.00**
BCA(Macro-BA@1)	47.69 ± 0.00	235.46 ± 0.01	23.56 ± 0.00	26.38 ± 0.00	**18.19 ± 0.00**	**59.09 ± 0.00**	**19.08 ± 0.00**	**14.96 ± 0.00**	**36.75 ± 0.00**
BCA(Macro-F_1@1)	52.68 ± 0.01	207.61 ± 0.06	24.44 ± 0.00	29.53 ± 0.01	13.77 ± 0.01	56.89 ± 0.00	17.28 ± 0.01	13.29 ± 0.01	*35.02 ± 0.02*
BCA(Macro-JS@1)	52.58 ± 0.01	205.07 ± 0.03	24.34 ± 0.00	*29.64 ± 0.01*	13.49 ± 0.01	56.75 ± 0.00	17.06 ± 0.00	13.14 ± 0.00	34.51 ± 0.01
BCA(Cov@1)	39.30 ± 0.01	188.91 ± 0.05	16.62 ± 0.01	23.42 ± 0.01	13.85 ± 0.01	56.93 ± 0.00	15.32 ± 0.01	11.50 ± 0.01	34.76 ± 0.01
FW(Macro-P@1)	55.34 ± 0.00	180.56 ± 0.02	24.48 ± 0.00	18.92 ± 0.01	9.27 ± 0.00	54.63 ± 0.00	11.25 ± 0.00	8.58 ± 0.00	23.53 ± 0.01
FW(Macro-R@1)	56.89 ± 0.00	*244.84 ± 0.00*	26.91 ± 0.00	24.33 ± 0.00	*16.31 ± 0.00*	*58.15 ± 0.00*	*17.70 ± 0.00*	*14.00 ± 0.00*	33.42 ± 0.00
FW(Macro-BA@1)	56.89 ± 0.00	**244.85 ± 0.00**	26.91 ± 0.00	24.33 ± 0.00	*16.31 ± 0.00*	*58.15 ± 0.00*	*17.70 ± 0.00*	*14.00 ± 0.00*	33.42 ± 0.00
FW(Macro-F_1@1)	53.17 ± 0.00	197.44 ± 0.00	24.16 ± 0.00	20.22 ± 0.00	11.19 ± 0.00	55.59 ± 0.00	13.07 ± 0.00	10.04 ± 0.00	27.02 ± 0.00
FW(Macro-JS@1)	54.60 ± 0.00	195.87 ± 0.00	24.77 ± 0.00	19.66 ± 0.00	10.68 ± 0.00	55.34 ± 0.00	12.60 ± 0.00	9.66 ± 0.00	26.11 ± 0.00

Method	Instance @1			Macro @1					
	P	PS	R	P	R	BA	F_1	JS	Cov
WikipediaLarge-500K									
Top-1	*66.83*	187.95	* 21.86	13.09	6.64	53.32	7.84	6.03	16.58
PS-1	**67.06**	241.60	**22.44**	19.81	10.98	55.49	12.63	9.96	24.66
Pow-$1_{\beta=0.25}$	65.55	232.96	*22.02*	18.49	10.07	55.03	11.67	9.13	23.29
Pow-$1_{\beta=0.5}$	63.46	256.87	21.64	21.49	12.49	56.25	14.17	11.21	27.34
Log-1	65.58	213.15	21.83	16.01	8.29	54.14	9.75	7.53	20.22
Macro-R_{prior}-1	58.19	275.15	20.07	24.03	15.15	57.57	16.58	13.17	31.56
Macro-BA_{prior}-1	58.19	275.15	20.07	24.03	15.15	57.57	16.58	13.17	31.56
BCA(Macro-P@1)	33.70 ± 0.00	153.83 ± 0.00	9.63 ± 0.00	**32.85 ± 0.00**	10.95 ± 0.00	55.47 ± 0.00	14.15 ± 0.00	10.84 ± 0.00	33.34 ± 0.00
BCA(Macro-R@1)	45.77 ± 0.00	260.53 ± 0.00	16.29 ± 0.00	27.17 ± 0.00	**17.17 ± 0.00**	**58.59 ± 0.00**	**18.50 ± 0.00**	**14.63 ± 0.00**	**35.68 ± 0.00**
BCA(Macro-BA@1)	45.77 ± 0.00	260.51 ± 0.00	16.29 ± 0.00	27.18 ± 0.00	**17.17 ± 0.00**	**58.59 ± 0.00**	**18.50 ± 0.00**	*14.62 ± 0.00*	**35.68 ± 0.00**
BCA(Macro-F_1@1)	50.85 ± 0.01	225.61 ± 0.10	17.05 ± 0.00	*30.96 ± 0.02*	13.19 ± 0.01	56.60 ± 0.00	16.74 ± 0.01	12.90 ± 0.01	35.11 ± 0.02
BCA(Macro-JS@1)	52.31 ± 0.02	230.21 ± 0.08	17.56 ± 0.01	30.24 ± 0.02	13.21 ± 0.01	56.60 ± 0.00	16.64 ± 0.01	12.88 ± 0.01	34.44 ± 0.02
BCA(Cov@1)	41.29 ± 0.02	213.29 ± 0.09	13.34 ± 0.00	25.54 ± 0.01	13.09 ± 0.01	56.54 ± 0.00	15.13 ± 0.01	11.39 ± 0.01	*35.18 ± 0.01*
FW(Macro-P@1)	59.00 ± 0.00	210.91 ± 0.00	18.96 ± 0.00	18.41 ± 0.00	9.26 ± 0.00	54.63 ± 0.00	10.97 ± 0.00	8.47 ± 0.00	22.95 ± 0.00
FW(Macro-R@1)	56.54 ± 0.00	**279.17 ± 0.00**	19.53 ± 0.00	24.89 ± 0.00	*15.95 ± 0.00*	*57.98 ± 0.00*	*17.28 ± 0.00*	13.77 ± 0.00	32.59 ± 0.00
FW(Macro-BA@1)	56.55 ± 0.00	*279.16 ± 0.00*	19.53 ± 0.00	24.89 ± 0.00	*15.95 ± 0.00*	*57.98 ± 0.00*	*17.28 ± 0.00*	13.77 ± 0.00	32.58 ± 0.00
FW(Macro-F_1@1)	50.37 ± 0.01	225.24 ± 0.03	16.09 ± 0.00	21.64 ± 0.00	11.89 ± 0.00	55.95 ± 0.00	13.52 ± 0.01	10.47 ± 0.00	28.36 ± 0.01
FW(Macro-JS@1)	50.60 ± 0.00	226.07 ± 0.00	16.16 ± 0.00	21.56 ± 0.00	11.95 ± 0.00	55.97 ± 0.00	13.46 ± 0.00	10.45 ± 0.00	28.06 ± 0.00
Amazon-670K									
Top-1	**44.97**	288.28	* *9.35*	4.78	3.08	51.54	3.42	2.81	5.82
PS-1	43.79	340.26	9.23	5.95	4.03	52.02	4.41	3.75	6.89
Pow-$1_{\beta=0.25}$	44.45	334.95	9.34	5.80	3.90	51.95	4.28	3.63	6.77
Pow-$1_{\beta=0.5}$	43.23	348.91	9.18	6.11	4.23	52.12	4.59	3.95	6.99
Log-1	*44.90*	314.25	**9.37**	5.34	3.49	51.75	3.86	3.22	6.37
Macro-R_{prior}-1	40.26	*351.00*	8.67	6.09	4.44	52.22	4.72	4.12	6.80
Macro-BA_{prior}-1	40.26	*351.00*	8.67	6.09	4.44	52.22	4.72	4.12	6.80
BCA(Macro-P@1)	40.31 ± 0.03	311.20 ± 0.26	8.42 ± 0.01	**8.46 ± 0.01**	3.79 ± 0.00	51.90 ± 0.00	4.69 ± 0.01	3.79 ± 0.00	**8.47 ± 0.01**
BCA(Macro-R@1)	38.88 ± 0.00	337.94 ± 0.00	8.37 ± 0.00	6.51 ± 0.00	**4.49 ± 0.00**	**52.25 ± 0.00**	4.92 ± 0.00	**4.31 ± 0.00**	6.92 ± 0.00
BCA(Macro-BA@1)	38.88 ± 0.00	337.94 ± 0.00	8.37 ± 0.00	6.51 ± 0.00	**4.49 ± 0.00**	**52.25 ± 0.00**	4.92 ± 0.00	**4.31 ± 0.00**	6.92 ± 0.00
BCA(Macro-F_1@1)	40.94 ± 0.03	330.86 ± 0.27	8.69 ± 0.01	*8.29 ± 0.01*	4.21 ± 0.00	52.11 ± 0.00	**5.06 ± 0.01**	4.21 ± 0.00	*8.37 ± 0.01*
BCA(Macro-JS@1)	41.22 ± 0.04	333.66 ± 0.32	8.74 ± 0.01	8.02 ± 0.01	4.23 ± 0.00	52.12 ± 0.00	*5.05 ± 0.01*	*4.23 ± 0.00*	8.13 ± 0.01
BCA(Cov@1)	41.01 ± 0.04	311.79 ± 0.30	8.35 ± 0.01	7.55 ± 0.01	3.68 ± 0.00	51.84 ± 0.00	4.44 ± 0.00	3.54 ± 0.00	*8.37 ± 0.01*
FW(Macro-P@1)	41.76 ± 0.05	301.02 ± 0.48	8.67 ± 0.01	6.01 ± 0.01	3.42 ± 0.01	51.71 ± 0.00	3.93 ± 0.01	3.21 ± 0.01	6.80 ± 0.01
FW(Macro-R@1)	39.63 ± 0.00	**351.28 ± 0.00**	8.58 ± 0.00	6.04 ± 0.00	*4.48 ± 0.00*	*52.24 ± 0.00*	4.73 ± 0.00	4.14 ± 0.00	6.73 ± 0.00
FW(Macro-BA@1)	39.63 ± 0.00	**351.28 ± 0.00**	8.58 ± 0.00	6.04 ± 0.00	*4.48 ± 0.00*	*52.24 ± 0.00*	4.73 ± 0.00	4.14 ± 0.00	6.73 ± 0.00
FW(Macro-F_1@1)	41.82 ± 0.03	318.95 ± 0.26	8.65 ± 0.01	6.17 ± 0.01	3.72 ± 0.00	51.86 ± 0.00	4.23 ± 0.01	3.49 ± 0.01	7.10 ± 0.01
FW(Macro-JS@1)	37.87 ± 0.00	327.13 ± 0.01	7.84 ± 0.00	5.48 ± 0.00	4.09 ± 0.00	52.04 ± 0.00	4.29 ± 0.00	3.72 ± 0.00	6.24 ± 0.00

Method	Instance @3 P	Instance @3 PS	Instance @3 R	Macro @3 P	Macro @3 R	Macro @3 BA	Macro @3 F_1	Macro @3 JS	Macro @3 Cov
RCV1x-2K									
Top-3	**72.18**	*78.69*	* **74.60**	13.79	4.65	52.31	5.44	3.76	26.14
PS-3	*71.39*	**78.92**	*74.02*	24.27	8.26	54.11	10.33	6.67	47.76
Pow-3$_{\beta=0.25}$	69.23	77.43	72.22	22.91	10.01	54.99	12.03	7.69	51.26
Pow-3$_{\beta=0.5}$	64.23	73.19	67.70	20.51	14.51	57.23	14.93	9.49	61.28
Log-3	69.38	77.35	72.29	21.94	8.63	54.29	10.56	6.80	45.48
Macro-R$_{\text{prior}}$-3	44.00	52.70	47.01	15.60	21.81	60.87	13.63	8.30	76.14
Macro-BA$_{\text{prior}}$-3	44.85	53.59	47.99	15.64	21.81	60.87	13.69	8.34	76.10
BCA(Macro-P@3)	42.75 ± 1.01	45.50 ± 1.09	42.65 ± 1.51	**36.29 ± 0.09**	2.92 ± 0.06	51.42 ± 0.03	3.57 ± 0.06	2.26 ± 0.05	38.62 ± 0.07
BCA(Macro-R@3)	22.89 ± 0.00	29.64 ± 0.00	22.33 ± 0.00	14.71 ± 0.00	*22.38 ± 0.00*	*61.14 ± 0.00*	11.27 ± 0.00	6.65 ± 0.00	*80.33 ± 0.00*
BCA(Macro-BA@3)	23.94 ± 0.00	30.75 ± 0.00	23.69 ± 0.00	14.73 ± 0.00	**22.39 ± 0.00**	**61.15 ± 0.00**	11.31 ± 0.00	6.68 ± 0.00	*80.33 ± 0.00*
BCA(Macro-F_1@3)	62.36 ± 0.01	70.67 ± 0.01	65.90 ± 0.01	23.69 ± 0.03	14.27 ± 0.04	57.11 ± 0.02	**16.58 ± 0.03**	**10.37 ± 0.03**	67.44 ± 0.07
BCA(Macro-JS@3)	62.99 ± 0.00	71.32 ± 0.00	66.55 ± 0.00	23.51 ± 0.02	13.92 ± 0.05	56.94 ± 0.02	*16.34 ± 0.02*	*10.27 ± 0.01*	66.19 ± 0.07
BCA(Cov@3)	2.04 ± 0.00	4.90 ± 0.00	0.71 ± 0.00	15.37 ± 0.08	16.95 ± 0.02	58.42 ± 0.01	5.41 ± 0.01	2.95 ± 0.01	**81.57 ± 0.07**
FW(Macro-P@3)	52.03 ± 0.00	57.90 ± 0.00	54.38 ± 0.00	*28.21 ± 0.01*	8.79 ± 0.00	54.36 ± 0.00	11.45 ± 0.00	7.05 ± 0.00	55.02 ± 0.04
FW(Macro-R@3)	43.75 ± 0.00	52.43 ± 0.00	46.76 ± 0.00	15.58 ± 0.00	21.91 ± 0.00	60.92 ± 0.00	13.61 ± 0.00	8.27 ± 0.00	76.34 ± 0.00
FW(Macro-BA@3)	44.60 ± 0.00	53.33 ± 0.00	47.73 ± 0.00	15.62 ± 0.00	21.90 ± 0.00	60.92 ± 0.00	13.66 ± 0.00	8.31 ± 0.00	76.30 ± 0.00
FW(Macro-F_1@3)	63.58 ± 0.00	71.67 ± 0.00	66.65 ± 0.00	24.31 ± 0.02	11.50 ± 0.01	55.73 ± 0.00	14.48 ± 0.01	9.17 ± 0.00	56.96 ± 0.03
FW(Macro-JS@3)	64.31 ± 0.00	72.43 ± 0.00	67.47 ± 0.00	24.08 ± 0.01	11.21 ± 0.02	55.58 ± 0.01	14.16 ± 0.00	9.05 ± 0.00	55.77 ± 0.02
EURLex-4K									
Top-3	**68.50**	157.08	* **40.52**	23.10	14.26	57.12	16.62	12.69	29.60
PS-3	67.48	*174.54*	39.93	27.66	20.04	60.01	21.92	17.28	36.76
Pow-3$_{\beta=0.25}$	67.88	172.65	40.13	27.18	19.08	59.53	21.19	16.66	35.71
Pow-3$_{\beta=0.5}$	64.85	**178.94**	38.35	28.53	22.32	61.15	23.53	18.59	39.47
Log-3	*68.23*	169.26	*40.32*	25.95	17.80	58.89	20.00	15.64	33.88
Macro-R$_{\text{prior}}$-3	49.88	162.26	29.42	27.95	24.91	62.44	23.58	18.26	43.03
Macro-BA$_{\text{prior}}$-3	50.04	162.56	29.51	28.00	24.92	62.44	23.62	18.30	43.00
BCA(Macro-P@3)	23.13 ± 0.16	71.98 ± 0.31	13.36 ± 0.12	**36.25 ± 0.04**	13.41 ± 0.04	56.68 ± 0.02	16.31 ± 0.03	12.78 ± 0.02	37.36 ± 0.05
BCA(Macro-R@3)	49.60 ± 0.00	161.78 ± 0.00	29.16 ± 0.00	27.76 ± 0.00	**25.47 ± 0.00**	**62.72 ± 0.00**	23.93 ± 0.00	18.54 ± 0.00	*43.75 ± 0.00*
BCA(Macro-BA@3)	49.67 ± 0.00	161.85 ± 0.00	29.21 ± 0.00	27.74 ± 0.00	*25.45 ± 0.00*	*62.70 ± 0.00*	23.93 ± 0.00	18.54 ± 0.00	43.73 ± 0.00
BCA(Macro-F_1@3)	60.35 ± 0.05	169.69 ± 0.15	35.69 ± 0.03	31.35 ± 0.03	22.76 ± 0.03	61.36 ± 0.02	**24.91 ± 0.04**	*19.56 ± 0.04*	42.35 ± 0.05
BCA(Macro-JS@3)	59.71 ± 0.04	164.96 ± 0.08	35.31 ± 0.03	*32.92 ± 0.07*	21.73 ± 0.05	60.85 ± 0.02	*24.87 ± 0.05*	**19.79 ± 0.05**	41.66 ± 0.08
BCA(Cov@3)	30.28 ± 0.08	116.90 ± 0.27	17.65 ± 0.08	27.87 ± 0.08	22.73 ± 0.04	61.34 ± 0.02	19.82 ± 0.03	14.67 ± 0.04	**45.86 ± 0.06**
FW(Macro-P@3)	59.59 ± 0.05	160.71 ± 0.16	35.13 ± 0.03	29.41 ± 0.07	19.94 ± 0.05	59.96 ± 0.02	21.81 ± 0.06	17.04 ± 0.06	37.60 ± 0.08
FW(Macro-R@3)	50.58 ± 0.00	163.72 ± 0.00	29.83 ± 0.00	28.09 ± 0.00	25.00 ± 0.00	62.48 ± 0.00	23.76 ± 0.00	18.45 ± 0.00	43.08 ± 0.00
FW(Macro-BA@3)	50.69 ± 0.00	163.84 ± 0.00	29.91 ± 0.00	28.07 ± 0.00	24.99 ± 0.00	62.48 ± 0.00	23.76 ± 0.00	18.45 ± 0.00	43.03 ± 0.00
FW(Macro-F_1@3)	61.94 ± 0.01	172.97 ± 0.02	36.56 ± 0.01	28.25 ± 0.00	21.84 ± 0.00	60.90 ± 0.00	23.03 ± 0.00	17.99 ± 0.00	39.42 ± 0.00
FW(Macro-JS@3)	62.95 ± 0.01	171.24 ± 0.02	37.18 ± 0.01	28.43 ± 0.00	20.95 ± 0.00	60.46 ± 0.00	22.53 ± 0.00	17.61 ± 0.00	38.67 ± 0.00

Method	Instance @3			Macro @3					
	P	PS	R	P	R	BA	F_1	JS	Cov
EURLex-4.3K									
Top-3	**81.21**	139.26	* **53.30**	22.55	14.15	57.07	16.41	12.93	27.09
PS-3	80.15	*152.03*	*52.62*	28.67	20.63	60.31	22.81	18.45	35.14
Pow-$3_{\beta=0.25}$	79.91	150.86	52.46	27.65	19.88	59.93	22.01	17.75	34.02
Pow-$3_{\beta=0.5}$	76.81	**153.65**	50.46	29.39	23.18	61.58	24.58	19.95	37.60
Log-3	*80.16*	148.64	52.59	26.50	18.66	59.32	20.84	16.79	32.33
Macro-R_{prior}-3	64.22	141.21	41.88	28.60	25.61	62.79	24.73	19.63	41.00
Macro-BA_{prior}-3	64.43	141.46	42.05	28.60	25.59	62.78	24.73	19.63	40.97
BCA(Macro-P@3)	34.71 ± 0.22	67.19 ± 0.25	22.08 ± 0.04	**36.47 ± 0.03**	13.19 ± 0.03	56.57 ± 0.02	15.93 ± 0.02	12.49 ± 0.02	37.57 ± 0.11
BCA(Macro-R@3)	63.94 ± 0.00	140.85 ± 0.00	41.68 ± 0.00	28.34 ± 0.00	**26.04 ± 0.00**	**63.01 ± 0.00**	24.94 ± 0.00	19.84 ± 0.00	*41.42 ± 0.00*
BCA(Macro-BA@3)	64.09 ± 0.00	141.02 ± 0.00	41.81 ± 0.00	28.34 ± 0.00	*26.02 ± 0.00*	*63.00 ± 0.00*	24.94 ± 0.00	19.83 ± 0.00	41.40 ± 0.00
BCA(Macro-F_1@3)	72.52 ± 0.02	144.52 ± 0.09	47.66 ± 0.02	*33.51 ± 0.07*	22.71 ± 0.07	61.34 ± 0.03	**25.81 ± 0.06**	**20.81 ± 0.05**	39.99 ± 0.06
BCA(Macro-JS@3)	73.43 ± 0.03	145.96 ± 0.10	48.19 ± 0.02	33.11 ± 0.06	22.61 ± 0.03	61.30 ± 0.02	*25.61 ± 0.04*	*20.73 ± 0.03*	39.54 ± 0.08
BCA(Cov@3)	22.11 ± 0.03	70.51 ± 0.07	13.49 ± 0.04	28.23 ± 0.06	21.68 ± 0.02	60.81 ± 0.01	18.37 ± 0.02	13.74 ± 0.02	**43.71 ± 0.05**
FW(Macro-P@3)	73.54 ± 0.03	143.78 ± 0.08	47.98 ± 0.02	30.51 ± 0.07	21.38 ± 0.03	60.68 ± 0.02	23.41 ± 0.04	18.72 ± 0.04	37.18 ± 0.04
FW(Macro-R@3)	64.89 ± 0.00	142.31 ± 0.00	42.34 ± 0.00	28.56 ± 0.00	25.66 ± 0.00	62.82 ± 0.00	24.78 ± 0.00	19.68 ± 0.00	40.95 ± 0.00
FW(Macro-BA@3)	65.04 ± 0.00	142.50 ± 0.00	42.45 ± 0.00	28.58 ± 0.00	25.66 ± 0.00	62.82 ± 0.00	24.79 ± 0.00	19.70 ± 0.00	40.95 ± 0.00
FW(Macro-F_1@3)	75.57 ± 0.19	147.85 ± 0.25	49.66 ± 0.13	28.69 ± 0.06	22.00 ± 0.16	60.99 ± 0.08	23.60 ± 0.06	18.93 ± 0.06	36.72 ± 0.08
FW(Macro-JS@3)	74.93 ± 0.00	148.82 ± 0.01	49.15 ± 0.00	29.23 ± 0.01	22.08 ± 0.00	61.03 ± 0.00	23.70 ± 0.00	19.07 ± 0.00	37.01 ± 0.01
AmazonCat-13K									
Top-3	**78.63**	93.42	* **59.53**	35.46	12.31	56.15	16.32	11.35	43.52
PS-3	*76.94*	**106.24**	*58.38*	57.15	42.37	71.18	45.90	36.09	76.79
Pow-$3_{\beta=0.25}$	73.02	*104.46*	55.25	54.72	43.59	71.79	46.25	36.39	75.76
Pow-$3_{\beta=0.5}$	65.43	100.85	49.11	49.91	54.14	77.07	49.45	38.54	82.45
Log-3	72.89	101.29	54.80	52.86	35.39	67.69	40.09	31.07	69.06
Macro-R_{prior}-3	47.33	83.83	33.96	35.94	62.69	81.34	41.24	30.03	87.63
Macro-BA_{prior}-3	47.71	84.23	34.39	35.97	62.69	81.34	41.27	30.06	87.63
BCA(Macro-P@3)	40.77 ± 0.12	46.86 ± 0.14	31.58 ± 0.11	**71.64 ± 0.02**	14.48 ± 0.01	57.23 ± 0.01	19.63 ± 0.01	14.31 ± 0.01	72.42 ± 0.01
BCA(Macro-R@3)	40.26 ± 0.00	75.43 ± 0.00	28.23 ± 0.00	34.01 ± 0.00	**63.64 ± 0.00**	*81.81 ± 0.00*	38.59 ± 0.00	27.96 ± 0.00	*89.16 ± 0.00*
BCA(Macro-BA@3)	40.62 ± 0.00	75.80 ± 0.00	28.66 ± 0.00	34.04 ± 0.00	**63.64 ± 0.00**	**81.82 ± 0.00**	38.61 ± 0.00	27.97 ± 0.00	*89.16 ± 0.00*
BCA(Macro-F_1@3)	66.94 ± 0.00	100.18 ± 0.00	50.45 ± 0.00	55.04 ± 0.01	52.30 ± 0.01	76.15 ± 0.00	**51.19 ± 0.01**	*40.10 ± 0.01*	84.71 ± 0.01
BCA(Macro-JS@3)	67.89 ± 0.00	98.84 ± 0.00	51.19 ± 0.00	61.52 ± 0.01	46.77 ± 0.01	73.38 ± 0.00	*51.05 ± 0.01*	**40.15 ± 0.01**	83.89 ± 0.01
BCA(Cov@3)	8.02 ± 0.01	26.41 ± 0.02	4.19 ± 0.00	28.10 ± 0.01	47.62 ± 0.02	73.80 ± 0.01	22.29 ± 0.01	14.60 ± 0.01	**89.45 ± 0.01**
FW(Macro-P@3)	55.76 ± 0.01	77.97 ± 0.01	41.81 ± 0.01	*62.13 ± 0.02*	36.86 ± 0.01	68.43 ± 0.01	42.77 ± 0.01	33.03 ± 0.01	77.24 ± 0.03
FW(Macro-R@3)	46.86 ± 0.00	83.30 ± 0.00	33.60 ± 0.00	35.46 ± 0.00	*62.95 ± 0.00*	81.47 ± 0.00	40.77 ± 0.00	29.58 ± 0.00	88.00 ± 0.00
FW(Macro-BA@3)	47.24 ± 0.00	83.69 ± 0.00	34.04 ± 0.00	35.49 ± 0.00	*62.95 ± 0.00*	81.47 ± 0.00	40.79 ± 0.00	29.60 ± 0.00	88.00 ± 0.00
FW(Macro-F_1@3)	67.84 ± 0.01	100.13 ± 0.01	50.97 ± 0.01	55.70 ± 0.02	48.73 ± 0.01	74.36 ± 0.01	49.51 ± 0.01	39.04 ± 0.01	80.56 ± 0.01
FW(Macro-JS@3)	68.34 ± 0.00	99.85 ± 0.00	51.33 ± 0.00	56.98 ± 0.00	46.68 ± 0.00	73.34 ± 0.00	48.90 ± 0.00	38.61 ± 0.00	79.52 ± 0.00

Method	Instance @3			Macro @3					
	P	PS	R	P	R	BA	F_1	JS	Cov
AmazonCat-14K									
Top-3	**69.10**	76.63	* **69.11**	39.74	15.50	57.75	20.09	13.61	54.89
PS-3	*68.18*	**81.39**	*68.38*	47.42	37.91	68.95	39.90	29.65	76.69
Pow-3$_{\beta=0.25}$	65.17	*80.73*	65.67	44.05	42.87	71.43	41.52	31.13	77.43
Pow-3$_{\beta=0.5}$	58.45	75.64	58.80	35.44	51.78	75.89	39.89	29.10	82.10
Log-3	66.28	80.30	66.38	46.34	35.93	67.96	38.39	28.64	72.45
Macro-R$_{\text{prior}}$-3	39.60	56.96	33.43	24.50	58.57	79.28	30.35	20.91	85.83
Macro-BA$_{\text{prior}}$-3	39.93	57.28	34.24	24.52	58.57	79.28	30.37	20.92	85.83
BCA(Macro-P@3)	34.18 ± 0.09	34.97 ± 0.01	40.16 ± 0.11	**65.01 ± 0.03**	8.58 ± 0.00	54.29 ± 0.00	12.62 ± 0.00	8.36 ± 0.00	70.21 ± 0.02
BCA(Macro-R@3)	36.10 ± 0.05	52.90 ± 0.05	29.03 ± 0.13	23.86 ± 0.00	**59.30 ± 0.00**	**79.65 ± 0.00**	28.87 ± 0.00	19.93 ± 0.00	*87.39 ± 0.00*
BCA(Macro-BA@3)	36.13 ± 0.07	52.91 ± 0.06	29.06 ± 0.13	23.86 ± 0.00	**59.30 ± 0.00**	**79.65 ± 0.00**	28.87 ± 0.00	19.93 ± 0.00	*87.39 ± 0.00*
BCA(Macro-F_1@3)	62.48 ± 0.00	77.48 ± 0.00	62.90 ± 0.00	48.27 ± 0.01	42.46 ± 0.00	71.22 ± 0.00	*43.37 ± 0.00*	*32.11 ± 0.00*	82.37 ± 0.01
BCA(Macro-JS@3)	62.57 ± 0.00	77.58 ± 0.00	63.07 ± 0.00	48.39 ± 0.00	42.33 ± 0.01	71.16 ± 0.00	**43.40 ± 0.00**	**32.17 ± 0.00**	82.20 ± 0.01
BCA(Cov@3)	3.77 ± 0.00	9.62 ± 0.00	2.49 ± 0.00	19.57 ± 0.00	37.18 ± 0.01	68.58 ± 0.01	11.30 ± 0.00	6.65 ± 0.00	**87.62 ± 0.00**
FW(Macro-P@3)	41.91 ± 0.00	48.78 ± 0.01	46.21 ± 0.00	*56.03 ± 0.04*	22.66 ± 0.02	61.32 ± 0.01	29.25 ± 0.02	20.00 ± 0.02	73.93 ± 0.05
FW(Macro-R@3)	38.67 ± 0.00	56.03 ± 0.00	31.78 ± 0.00	24.12 ± 0.00	*58.97 ± 0.00*	*79.48 ± 0.00*	29.86 ± 0.00	20.53 ± 0.00	86.38 ± 0.00
FW(Macro-BA@3)	39.10 ± 0.00	56.46 ± 0.00	32.87 ± 0.00	24.14 ± 0.00	*58.97 ± 0.00*	*79.48 ± 0.00*	29.88 ± 0.00	20.54 ± 0.00	86.38 ± 0.00
FW(Macro-F_1@3)	62.98 ± 0.00	77.09 ± 0.02	63.33 ± 0.00	47.29 ± 0.00	42.71 ± 0.00	71.35 ± 0.00	42.72 ± 0.00	31.78 ± 0.00	79.39 ± 0.03
FW(Macro-JS@3)	62.66 ± 0.00	76.74 ± 0.05	63.08 ± 0.01	47.51 ± 0.00	42.37 ± 0.00	71.18 ± 0.00	42.70 ± 0.00	31.82 ± 0.00	79.14 ± 0.00
Wiki10-31K									
Top-3	**72.46**	104.98	* **12.58**	2.57	0.48	50.24	0.73	0.47	2.86
PS-3	*69.78*	150.53	*12.24*	6.82	2.75	51.37	3.47	2.66	7.80
Pow-3$_{\beta=0.25}$	69.25	150.25	12.12	6.40	2.49	51.24	3.19	2.40	7.44
Pow-3$_{\beta=0.5}$	61.20	173.63	10.74	8.69	4.35	52.17	5.19	4.11	10.34
Log-3	65.46	143.17	11.39	5.72	2.10	51.05	2.76	2.01	6.82
Macro-R$_{\text{prior}}$-3	43.30	*181.78*	7.52	10.82	**6.67**	**53.33**	**7.26**	**5.90**	13.62
Macro-BA$_{\text{prior}}$-3	43.52	**181.88**	7.56	10.83	**6.67**	**53.33**	**7.26**	**5.90**	13.61
BCA(Macro-P@3)	38.10 ± 0.09	126.65 ± 0.43	6.52 ± 0.02	**12.62 ± 0.04**	4.70 ± 0.04	52.34 ± 0.02	5.82 ± 0.04	4.69 ± 0.04	12.70 ± 0.05
BCA(Macro-R@3)	30.44 ± 0.00	150.77 ± 0.00	5.21 ± 0.00	10.90 ± 0.00	6.19 ± 0.00	53.09 ± 0.00	6.96 ± 0.00	5.62 ± 0.00	*13.69 ± 0.00*
BCA(Macro-BA@3)	30.50 ± 0.00	150.88 ± 0.00	5.22 ± 0.00	10.91 ± 0.00	6.19 ± 0.00	53.09 ± 0.00	6.97 ± 0.00	5.62 ± 0.00	**13.70 ± 0.00**
BCA(Macro-F_1@3)	42.55 ± 0.02	150.21 ± 0.38	7.37 ± 0.00	*11.97 ± 0.06*	5.03 ± 0.03	52.51 ± 0.02	6.28 ± 0.04	4.99 ± 0.03	13.24 ± 0.06
BCA(Macro-JS@3)	43.85 ± 0.03	152.57 ± 0.32	7.60 ± 0.00	11.76 ± 0.04	4.96 ± 0.03	52.48 ± 0.01	6.21 ± 0.03	4.92 ± 0.03	13.09 ± 0.03
BCA(Cov@3)	33.60 ± 0.04	139.66 ± 0.17	5.73 ± 0.01	9.90 ± 0.00	4.69 ± 0.01	52.34 ± 0.00	5.55 ± 0.01	4.21 ± 0.01	13.40 ± 0.01
FW(Macro-P@3)	38.26 ± 0.33	95.62 ± 1.17	6.54 ± 0.06	5.50 ± 0.07	2.69 ± 0.06	51.34 ± 0.03	3.07 ± 0.05	2.44 ± 0.05	6.91 ± 0.09
FW(Macro-R@3)	43.26 ± 0.00	181.71 ± 0.00	7.51 ± 0.00	10.81 ± 0.00	**6.67 ± 0.00**	**53.33 ± 0.00**	**7.26 ± 0.00**	**5.90 ± 0.00**	13.61 ± 0.00
FW(Macro-BA@3)	43.52 ± 0.00	**181.88 ± 0.00**	7.56 ± 0.00	10.83 ± 0.00	**6.67 ± 0.00**	**53.33 ± 0.00**	**7.26 ± 0.00**	**5.90 ± 0.00**	13.61 ± 0.00
FW(Macro-F_1@3)	41.41 ± 0.20	154.83 ± 1.29	7.13 ± 0.03	9.26 ± 0.07	5.05 ± 0.05	52.52 ± 0.03	5.75 ± 0.05	4.56 ± 0.05	11.94 ± 0.08
FW(Macro-JS@3)	39.00 ± 0.00	168.39 ± 0.02	6.72 ± 0.00	10.79 ± 0.00	*6.49 ± 0.00*	*53.24 ± 0.00*	*7.00 ± 0.00*	*5.69 ± 0.00*	13.49 ± 0.00

Method	Instance @3			Macro @3					
	P	PS	R	P	R	BA	F_1	JS	Cov
WikiLSHTC-325K									
Top-3	*41.94*	118.44	* 47.00	18.61	15.71	57.85	15.19	11.59	29.71
PS-3	**43.28**	135.38	**48.99**	21.42	20.35	60.18	18.83	14.45	36.20
Pow-$3_{\beta=0.25}$	41.90	132.25	*47.86*	20.74	19.48	59.74	18.13	13.89	34.97
Pow-$3_{\beta=0.5}$	40.47	140.67	47.11	21.78	22.67	61.33	20.19	15.43	39.20
Log-3	41.55	125.38	47.20	19.66	17.49	58.74	16.62	12.70	32.23
Macro-R_{prior}-3	37.08	*145.82*	44.57	21.40	26.31	63.16	21.40	16.00	44.12
Macro-BA_{prior}-3	37.08	*145.82*	44.57	21.40	26.31	63.16	21.40	16.00	44.12
BCA(Macro-P@3)	15.22 ± 0.03	53.10 ± 0.04	15.10 ± 0.03	**34.07 ± 0.01**	12.63 ± 0.01	56.32 ± 0.00	15.38 ± 0.00	11.86 ± 0.00	35.60 ± 0.01
BCA(Macro-R@3)	29.31 ± 0.00	135.03 ± 0.00	37.44 ± 0.00	22.22 ± 0.00	**28.55 ± 0.00**	**64.27 ± 0.00**	21.82 ± 0.00	16.10 ± 0.00	**49.01 ± 0.00**
BCA(Macro-BA@3)	29.31 ± 0.00	135.02 ± 0.00	37.44 ± 0.00	22.22 ± 0.00	**28.55 ± 0.00**	**64.27 ± 0.00**	21.82 ± 0.00	16.10 ± 0.00	**49.01 ± 0.00**
BCA(Macro-F_1@3)	34.74 ± 0.00	127.05 ± 0.01	40.20 ± 0.00	27.95 ± 0.00	23.39 ± 0.00	61.70 ± 0.00	**23.23 ± 0.00**	*17.86 ± 0.00*	45.30 ± 0.01
BCA(Macro-JS@3)	34.23 ± 0.00	121.71 ± 0.01	39.24 ± 0.00	*29.51 ± 0.01*	22.17 ± 0.00	61.08 ± 0.00	*23.12 ± 0.00*	**17.87 ± 0.00**	44.36 ± 0.01
BCA(Cov@3)	21.72 ± 0.01	110.05 ± 0.01	25.19 ± 0.01	21.74 ± 0.00	24.61 ± 0.00	62.31 ± 0.00	18.97 ± 0.00	13.86 ± 0.00	*46.48 ± 0.01*
FW(Macro-P@3)	31.89 ± 0.01	112.54 ± 0.07	37.11 ± 0.01	24.37 ± 0.00	19.40 ± 0.01	59.70 ± 0.00	19.30 ± 0.00	14.91 ± 0.00	37.61 ± 0.02
FW(Macro-R@3)	36.23 ± 0.00	**146.30 ± 0.00**	43.74 ± 0.00	21.84 ± 0.00	*27.16 ± 0.00*	*63.58 ± 0.00*	21.89 ± 0.00	16.31 ± 0.00	45.36 ± 0.00
FW(Macro-BA@3)	36.24 ± 0.00	**146.30 ± 0.00**	43.74 ± 0.00	21.84 ± 0.00	*27.16 ± 0.00*	*63.58 ± 0.00*	21.89 ± 0.00	16.31 ± 0.00	45.36 ± 0.00
FW(Macro-F_1@3)	36.60 ± 0.00	128.24 ± 0.01	42.34 ± 0.00	23.24 ± 0.00	21.30 ± 0.00	60.65 ± 0.00	20.27 ± 0.00	15.60 ± 0.00	39.13 ± 0.00
FW(Macro-JS@3)	37.18 ± 0.01	128.47 ± 0.04	42.71 ± 0.01	23.05 ± 0.00	21.17 ± 0.00	60.58 ± 0.00	20.09 ± 0.00	15.46 ± 0.00	38.91 ± 0.00
WikipediaLarge-500K									
Top-3	*47.79*	142.69	* 39.77	19.92	16.16	58.08	15.94	12.35	30.83
PS-3	**48.38**	164.30	**40.94**	23.94	21.31	60.66	20.28	15.79	38.30
Pow-$3_{\beta=0.25}$	47.49	160.34	*40.31*	23.06	20.18	60.09	19.38	15.07	36.77
Pow-$3_{\beta=0.5}$	45.52	171.09	39.48	24.89	23.53	61.77	21.85	16.96	41.46
Log-3	47.50	151.45	39.96	21.40	17.93	58.96	17.50	13.57	33.56
Macro-R_{prior}-3	40.63	176.57	36.30	25.36	27.11	63.55	23.56	18.00	46.58
Macro-BA_{prior}-3	40.64	176.58	36.30	25.36	27.11	63.55	23.56	18.00	46.58
BCA(Macro-P@3)	18.64 ± 0.04	68.89 ± 0.08	13.54 ± 0.03	**36.20 ± 0.01**	13.72 ± 0.01	56.86 ± 0.00	16.67 ± 0.00	13.03 ± 0.00	37.63 ± 0.01
BCA(Macro-R@3)	31.61 ± 0.00	163.01 ± 0.00	29.59 ± 0.00	26.81 ± 0.00	**29.23 ± 0.00**	**64.61 ± 0.00**	**24.44 ± 0.00**	18.40 ± 0.00	**51.73 ± 0.00**
BCA(Macro-BA@3)	31.61 ± 0.00	163.00 ± 0.00	29.59 ± 0.00	26.81 ± 0.00	**29.23 ± 0.00**	**64.61 ± 0.00**	**24.44 ± 0.00**	18.40 ± 0.00	**51.73 ± 0.00**
BCA(Macro-F_1@3)	37.98 ± 0.00	148.08 ± 0.00	32.01 ± 0.00	*31.50 ± 0.00*	22.99 ± 0.00	61.49 ± 0.00	*24.15 ± 0.00*	**18.80 ± 0.00**	46.38 ± 0.00
BCA(Macro-JS@3)	38.43 ± 0.00	148.42 ± 0.00	32.22 ± 0.00	31.19 ± 0.00	22.73 ± 0.00	61.37 ± 0.00	23.91 ± 0.00	*18.63 ± 0.00*	45.85 ± 0.00
BCA(Cov@3)	24.55 ± 0.00	134.12 ± 0.01	21.64 ± 0.00	26.23 ± 0.00	25.26 ± 0.00	62.63 ± 0.00	21.59 ± 0.00	16.08 ± 0.00	*50.06 ± 0.01*
FW(Macro-P@3)	37.23 ± 0.00	137.13 ± 0.00	30.29 ± 0.00	25.67 ± 0.00	19.89 ± 0.00	59.94 ± 0.00	19.68 ± 0.00	15.32 ± 0.00	38.09 ± 0.00
FW(Macro-R@3)	39.60 ± 0.00	*177.57 ± 0.00*	35.49 ± 0.00	25.87 ± 0.00	*28.08 ± 0.00*	*64.04 ± 0.00*	24.05 ± 0.00	18.30 ± 0.00	47.91 ± 0.00
FW(Macro-BA@3)	39.61 ± 0.00	**177.58 ± 0.00**	35.50 ± 0.00	25.87 ± 0.00	*28.08 ± 0.00*	*64.04 ± 0.00*	24.05 ± 0.00	18.30 ± 0.00	47.91 ± 0.00
FW(Macro-F_1@3)	38.52 ± 0.00	150.09 ± 0.00	32.79 ± 0.00	25.19 ± 0.00	21.45 ± 0.00	60.73 ± 0.00	20.86 ± 0.00	16.23 ± 0.00	40.55 ± 0.00
FW(Macro-JS@3)	39.15 ± 0.00	149.00 ± 0.00	32.96 ± 0.00	24.94 ± 0.00	20.99 ± 0.00	60.49 ± 0.00	20.49 ± 0.00	15.95 ± 0.00	39.88 ± 0.00

Method	Instance @3 P	Instance @3 PS	Instance @3 R	Macro @3 P	Macro @3 R	Macro @3 BA	Macro @3 F_1	Macro @3 JS	Macro @3 Cov
Amazon-670K									
Top-3	*40.20*	273.84	* 23.31	10.46	9.04	54.52	9.01	7.60	13.78
PS-3	39.87	303.17	23.26	12.27	10.62	55.31	10.62	9.08	15.84
Pow-$3_{\beta=0.25}$	40.18	299.30	**23.42**	11.99	10.35	55.17	10.38	8.87	15.48
Pow-$3_{\beta=0.5}$	39.47	**307.72**	23.10	12.66	10.96	55.48	10.96	9.40	16.20
Log-3	**40.30**	287.86	*23.41*	11.22	9.70	54.85	9.70	8.24	14.64
Macro-R_{prior}-3	37.20	*307.64*	21.94	13.03	11.35	55.68	11.22	9.64	16.39
Macro-BA_{prior}-3	37.20	*307.64*	21.94	13.03	11.35	55.68	11.22	9.64	16.39
BCA(Macro-P@3)	30.23 ± 0.01	251.23 ± 0.18	17.86 ± 0.01	**17.23 ± 0.01**	10.09 ± 0.01	55.04 ± 0.01	11.69 ± 0.01	10.07 ± 0.01	17.31 ± 0.01
BCA(Macro-R@3)	35.80 ± 0.00	298.22 ± 0.00	21.14 ± 0.00	14.35 ± 0.00	**11.44 ± 0.00**	**55.72 ± 0.00**	11.80 ± 0.00	10.21 ± 0.00	16.93 ± 0.00
BCA(Macro-BA@3)	35.80 ± 0.00	298.22 ± 0.00	21.14 ± 0.00	14.35 ± 0.00	**11.44 ± 0.00**	**55.72 ± 0.00**	11.80 ± 0.00	10.21 ± 0.00	16.93 ± 0.00
BCA(Macro-F_1@3)	35.13 ± 0.01	280.38 ± 0.06	20.69 ± 0.01	*16.89 ± 0.00*	10.68 ± 0.00	55.34 ± 0.00	**12.22 ± 0.00**	*10.54 ± 0.00*	**17.71 ± 0.00**
BCA(Macro-JS@3)	35.44 ± 0.01	282.83 ± 0.09	20.86 ± 0.01	16.63 ± 0.01	10.71 ± 0.00	55.36 ± 0.00	*12.17 ± 0.01*	**10.56 ± 0.01**	17.49 ± 0.01
BCA(Cov@3)	34.48 ± 0.01	281.68 ± 0.07	19.90 ± 0.01	14.03 ± 0.00	10.52 ± 0.00	55.26 ± 0.00	11.08 ± 0.00	9.24 ± 0.00	*17.57 ± 0.00*
FW(Macro-P@3)	36.30 ± 0.01	273.78 ± 0.12	21.12 ± 0.01	12.22 ± 0.01	9.57 ± 0.00	54.78 ± 0.00	9.89 ± 0.00	8.39 ± 0.00	14.74 ± 0.01
FW(Macro-R@3)	36.68 ± 0.00	306.96 ± 0.00	21.68 ± 0.00	13.06 ± 0.00	*11.40 ± 0.00*	*55.70 ± 0.00*	11.23 ± 0.00	9.65 ± 0.00	16.35 ± 0.00
FW(Macro-BA@3)	36.68 ± 0.00	306.96 ± 0.00	21.68 ± 0.00	13.06 ± 0.00	*11.40 ± 0.00*	*55.70 ± 0.00*	11.23 ± 0.00	9.65 ± 0.00	16.35 ± 0.00
FW(Macro-F_1@3)	37.35 ± 0.01	283.53 ± 0.08	21.67 ± 0.01	12.12 ± 0.00	9.87 ± 0.00	54.94 ± 0.00	10.03 ± 0.01	8.52 ± 0.01	15.09 ± 0.01
FW(Macro-JS@3)	37.56 ± 0.00	281.88 ± 0.00	21.79 ± 0.00	11.90 ± 0.00	9.75 ± 0.00	54.87 ± 0.00	9.88 ± 0.00	8.40 ± 0.00	14.78 ± 0.00

Method	Instance @5 P	Instance @5 PS	Instance @5 R	Macro @5 P	Macro @5 R	Macro @5 BA	Macro @5 F_1	Macro @5 JS	Macro @5 Cov
RCV1x-2K									
Top-5	**51.67**	*56.92*	* **81.17**	13.37	7.61	53.76	7.62	4.95	35.91
PS-5	*51.23*	**57.21**	*80.69*	19.38	12.18	56.04	12.33	7.83	53.99
Pow-$5_{\beta=0.25}$	50.17	56.63	79.43	18.37	14.16	57.03	13.52	8.54	56.26
Pow-$5_{\beta=0.5}$	47.40	54.47	76.03	16.57	19.80	59.85	15.56	9.87	65.34
Log-5	50.31	56.59	79.50	17.96	12.57	56.23	12.23	7.76	50.95
Macro-R_{prior}-5	34.89	41.94	57.92	14.00	29.11	64.49	14.25	8.75	80.09
Macro-BA_{prior}-5	35.35	42.43	58.73	14.03	29.09	64.48	14.29	8.78	80.09
BCA(Macro-P@5)	28.43 ± 0.13	30.36 ± 0.13	45.70 ± 0.47	**36.36 ± 0.04**	3.30 ± 0.04	51.57 ± 0.02	3.70 ± 0.04	2.35 ± 0.03	39.20 ± 0.15
BCA(Macro-R@5)	29.13 ± 0.00	35.64 ± 0.00	48.63 ± 0.00	13.62 ± 0.00	*29.81 ± 0.00*	*64.83 ± 0.00*	13.18 ± 0.00	8.04 ± 0.00	*81.15 ± 0.00*
BCA(Macro-BA@5)	29.78 ± 0.00	36.33 ± 0.00	49.82 ± 0.00	13.65 ± 0.00	**29.82 ± 0.00**	**64.84 ± 0.00**	13.22 ± 0.00	8.07 ± 0.00	*81.15 ± 0.00*
BCA(Macro-F_1@5)	45.52 ± 0.00	51.70 ± 0.00	73.47 ± 0.00	23.67 ± 0.04	17.55 ± 0.02	58.72 ± 0.01	**18.67 ± 0.03**	*11.84 ± 0.02*	70.46 ± 0.08
BCA(Macro-JS@5)	45.69 ± 0.00	51.88 ± 0.00	73.69 ± 0.00	23.67 ± 0.01	17.40 ± 0.02	58.64 ± 0.01	*18.60 ± 0.01*	**11.86 ± 0.01**	69.84 ± 0.09
BCA(Cov@5)	1.70 ± 0.00	4.01 ± 0.00	0.96 ± 0.00	12.90 ± 0.06	22.05 ± 0.03	60.92 ± 0.02	5.21 ± 0.01	2.82 ± 0.01	**84.63 ± 0.02**
FW(Macro-P@5)	35.32 ± 0.00	39.16 ± 0.00	57.07 ± 0.01	*29.04 ± 0.01*	8.77 ± 0.00	54.32 ± 0.00	11.69 ± 0.00	7.24 ± 0.00	53.50 ± 0.03
FW(Macro-R@5)	34.78 ± 0.00	41.82 ± 0.00	57.75 ± 0.00	14.00 ± 0.00	29.11 ± 0.00	64.49 ± 0.00	14.23 ± 0.00	8.73 ± 0.00	80.21 ± 0.00
FW(Macro-BA@5)	35.26 ± 0.00	42.32 ± 0.00	58.59 ± 0.00	14.04 ± 0.00	29.10 ± 0.00	64.48 ± 0.00	14.28 ± 0.00	8.77 ± 0.00	80.21 ± 0.00
FW(Macro-F_1@5)	46.05 ± 0.00	52.22 ± 0.00	73.71 ± 0.00	23.12 ± 0.01	15.51 ± 0.01	57.70 ± 0.00	17.20 ± 0.01	11.09 ± 0.00	61.55 ± 0.02
FW(Macro-JS@5)	46.30 ± 0.00	52.44 ± 0.00	74.05 ± 0.00	22.77 ± 0.01	14.82 ± 0.00	57.35 ± 0.00	16.60 ± 0.00	10.73 ± 0.00	60.02 ± 0.02

Method	Instance @5			Macro @5					
	P	PS	R	P	R	BA	F_1	JS	Cov
EURLex-4K									
Top-5	57.47	135.71	* 55.66	25.30	22.05	61.00	22.41	17.65	36.71
PS-5	57.36	*144.55*	55.56	27.14	26.73	63.34	25.56	20.20	41.75
Pow-$5_{\beta=0.25}$	*57.48*	142.87	*55.67*	26.60	25.67	62.81	24.85	19.62	40.57
Pow-$5_{\beta=0.5}$	55.40	**145.34**	53.72	26.90	28.47	64.21	26.19	20.60	43.45
Log-5	**57.57**	141.35	**55.75**	26.16	24.76	62.35	24.20	19.11	39.44
Macro-R_{prior}-5	43.99	131.81	42.81	25.27	30.73	65.33	25.19	18.98	46.93
Macro-BA_{prior}-5	44.09	131.97	42.91	25.27	30.74	65.34	25.21	19.00	46.93
BCA(Macro-P@5)	16.88 ± 0.36	48.06 ± 0.72	16.13 ± 0.34	**36.26 ± 0.03**	13.93 ± 0.10	56.91 ± 0.05	16.26 ± 0.06	12.72 ± 0.05	37.94 ± 0.07
BCA(Macro-R@5)	43.29 ± 0.00	131.10 ± 0.00	42.14 ± 0.00	25.07 ± 0.00	**31.32 ± 0.00**	**65.62 ± 0.00**	25.34 ± 0.00	19.17 ± 0.00	*47.66 ± 0.00*
BCA(Macro-BA@5)	43.39 ± 0.00	131.24 ± 0.00	42.23 ± 0.00	25.07 ± 0.00	**31.32 ± 0.00**	**65.62 ± 0.00**	25.35 ± 0.00	19.18 ± 0.00	*47.66 ± 0.00*
BCA(Macro-F_1@5)	50.38 ± 0.02	133.52 ± 0.08	48.85 ± 0.02	*30.46 ± 0.02*	27.68 ± 0.04	63.81 ± 0.02	**27.36 ± 0.02**	**21.46 ± 0.02**	45.61 ± 0.05
BCA(Macro-JS@5)	51.27 ± 0.02	134.54 ± 0.08	49.63 ± 0.02	30.41 ± 0.04	27.48 ± 0.06	63.71 ± 0.03	*27.23 ± 0.04*	*21.41 ± 0.03*	45.32 ± 0.08
BCA(Cov@5)	21.33 ± 0.08	83.39 ± 0.21	20.57 ± 0.10	25.48 ± 0.11	27.05 ± 0.07	63.48 ± 0.04	19.40 ± 0.06	13.83 ± 0.05	**49.21 ± 0.08**
FW(Macro-P@5)	43.39 ± 0.00	113.02 ± 0.01	42.20 ± 0.00	**29.62 ± 0.01**	22.63 ± 0.00	61.28 ± 0.00	23.01 ± 0.00	17.90 ± 0.00	40.15 ± 0.01
FW(Macro-R@5)	46.78 ± 0.00	136.67 ± 0.00	45.49 ± 0.00	25.14 ± 0.00	*30.91 ± 0.00*	*65.42 ± 0.00*	25.49 ± 0.00	19.37 ± 0.00	46.56 ± 0.00
FW(Macro-BA@5)	46.89 ± 0.00	136.83 ± 0.00	45.61 ± 0.00	25.15 ± 0.00	*30.91 ± 0.00*	*65.42 ± 0.00*	25.50 ± 0.00	19.39 ± 0.00	46.56 ± 0.00
FW(Macro-F_1@5)	52.52 ± 0.00	139.16 ± 0.01	50.92 ± 0.00	27.92 ± 0.01	28.05 ± 0.00	63.99 ± 0.00	**26.07 ± 0.00**	20.34 ± 0.00	44.24 ± 0.01
FW(Macro-JS@5)	53.82 ± 0.00	140.96 ± 0.01	52.11 ± 0.00	27.09 ± 0.00	27.75 ± 0.00	63.85 ± 0.00	25.69 ± 0.00	**19.98 ± 0.00**	43.75 ± 0.01
EURLex-4.3K									
Top-5	**68.50**	120.68	* **71.61**	25.36	22.57	61.26	22.73	18.31	34.54
PS-5	*68.40*	*125.92*	*71.55*	27.67	26.79	63.37	26.04	21.19	38.91
Pow-$5_{\beta=0.25}$	68.31	125.22	71.46	26.98	26.13	63.05	25.39	20.64	38.00
Pow-$5_{\beta=0.5}$	66.58	**126.40**	69.83	27.31	28.84	64.40	26.79	21.68	40.62
Log-5	68.37	124.01	71.52	26.25	25.05	62.51	24.54	19.93	36.64
Macro-R_{prior}-5	57.69	117.55	60.74	26.10	31.35	65.65	26.37	20.73	43.74
Macro-BA_{prior}-5	57.84	117.69	60.90	26.10	31.30	65.63	26.37	20.74	43.71
BCA(Macro-P@5)	26.08 ± 0.23	47.89 ± 0.26	27.05 ± 0.26	**36.64 ± 0.05**	14.09 ± 0.04	57.00 ± 0.02	16.48 ± 0.03	12.97 ± 0.03	38.27 ± 0.04
BCA(Macro-R@5)	57.21 ± 0.00	116.86 ± 0.00	60.26 ± 0.00	25.70 ± 0.00	**31.66 ± 0.00**	*65.80 ± 0.00*	26.25 ± 0.00	20.70 ± 0.00	*43.85 ± 0.00*
BCA(Macro-BA@5)	57.42 ± 0.00	117.13 ± 0.00	60.49 ± 0.00	25.72 ± 0.00	**31.66 ± 0.00**	**65.81 ± 0.00**	26.27 ± 0.00	20.71 ± 0.00	*43.85 ± 0.00*
BCA(Macro-F_1@5)	62.38 ± 0.02	118.51 ± 0.07	65.18 ± 0.02	31.25 ± 0.04	28.23 ± 0.05	64.09 ± 0.02	**28.22 ± 0.04**	**22.99 ± 0.04**	42.51 ± 0.05
BCA(Macro-JS@5)	61.70 ± 0.02	114.87 ± 0.07	64.44 ± 0.02	*33.21 ± 0.04*	26.02 ± 0.06	62.99 ± 0.03	*27.87 ± 0.04*	*22.89 ± 0.04*	41.34 ± 0.06
BCA(Cov@5)	15.26 ± 0.04	49.12 ± 0.09	15.29 ± 0.05	26.18 ± 0.05	25.11 ± 0.04	62.50 ± 0.02	18.02 ± 0.02	13.22 ± 0.02	**45.83 ± 0.05**
FW(Macro-P@5)	57.05 ± 0.00	104.01 ± 0.01	59.83 ± 0.00	**30.92 ± 0.00**	22.27 ± 0.00	61.11 ± 0.00	24.13 ± 0.00	19.54 ± 0.00	37.25 ± 0.01
FW(Macro-R@5)	58.16 ± 0.00	118.31 ± 0.00	61.21 ± 0.00	26.07 ± 0.00	*31.43 ± 0.00*	65.69 ± 0.00	26.42 ± 0.00	20.79 ± 0.00	43.71 ± 0.00
FW(Macro-BA@5)	58.30 ± 0.00	118.48 ± 0.00	61.36 ± 0.00	26.08 ± 0.00	31.42 ± 0.00	65.69 ± 0.00	26.44 ± 0.00	20.80 ± 0.00	43.69 ± 0.00
FW(Macro-F_1@5)	64.13 ± 0.01	120.41 ± 0.03	67.01 ± 0.01	29.46 ± 0.02	27.16 ± 0.02	63.56 ± 0.01	26.85 ± 0.02	21.86 ± 0.02	40.04 ± 0.01
FW(Macro-JS@5)	63.34 ± 0.00	117.63 ± 0.00	66.11 ± 0.00	30.02 ± 0.00	25.80 ± 0.00	62.88 ± 0.00	26.33 ± 0.00	21.57 ± 0.00	39.12 ± 0.00

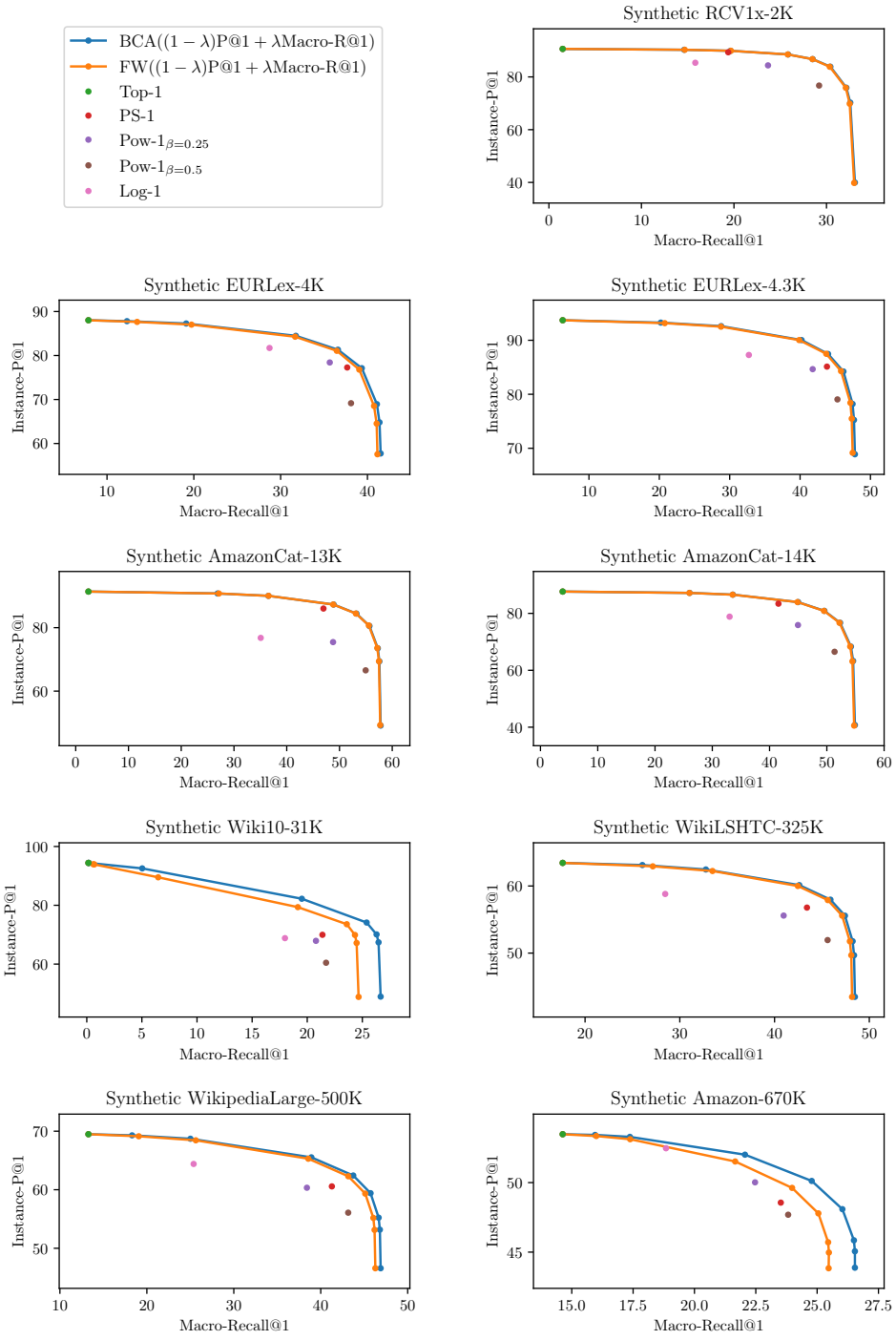
Method	Instance @5			Macro @5					
	P	PS	R	P	R	BA	F_1	JS	Cov
AmazonCat-13K									
Top-5	**63.96**	81.53	* **74.90**	46.66	32.98	66.48	35.32	27.15	63.81
PS-5	*63.67*	**85.66**	*74.68*	51.29	50.35	75.17	47.74	37.22	79.93
Pow-5$_{\beta=0.25}$	61.78	*84.76*	72.76	49.36	51.36	75.67	47.67	37.05	79.23
Pow-5$_{\beta=0.5}$	56.45	81.66	66.76	41.76	61.10	80.54	46.77	35.18	84.90
Log-5	61.51	83.17	72.20	50.12	45.50	72.74	44.99	35.04	74.83
Macro-R$_{\text{prior}}$-5	39.62	65.19	46.13	27.37	68.86	84.42	34.90	24.33	88.88
Macro-BA$_{\text{prior}}$-5	39.99	65.58	46.75	27.40	68.86	84.42	34.93	24.36	88.88
BCA(Macro-P@5)	31.70 ± 0.11	36.00 ± 0.12	38.65 ± 0.16	**71.42 ± 0.02**	14.83 ± 0.01	57.40 ± 0.01	19.79 ± 0.02	14.40 ± 0.02	72.59 ± 0.03
BCA(Macro-R@5)	35.39 ± 0.00	60.42 ± 0.00	40.97 ± 0.00	27.06 ± 0.00	**69.39 ± 0.00**	**84.68 ± 0.00**	33.97 ± 0.00	23.89 ± 0.00	*89.35 ± 0.00*
BCA(Macro-BA@5)	35.75 ± 0.00	60.80 ± 0.00	41.60 ± 0.00	27.08 ± 0.00	**69.39 ± 0.00**	**84.68 ± 0.00**	33.99 ± 0.00	23.91 ± 0.00	*89.35 ± 0.00*
BCA(Macro-F_1@5)	56.70 ± 0.02	77.43 ± 0.02	66.88 ± 0.03	60.91 ± 0.02	49.11 ± 0.01	74.54 ± 0.01	**52.15 ± 0.02**	*41.08 ± 0.01*	84.85 ± 0.02
BCA(Macro-JS@5)	56.56 ± 0.02	77.29 ± 0.02	66.65 ± 0.03	*60.95 ± 0.01*	49.07 ± 0.01	74.53 ± 0.01	**52.15 ± 0.01**	**41.11 ± 0.01**	84.79 ± 0.01
BCA(Cov@5)	5.96 ± 0.00	18.58 ± 0.01	5.09 ± 0.00	21.13 ± 0.03	51.96 ± 0.01	75.96 ± 0.01	19.19 ± 0.01	12.22 ± 0.01	**89.82 ± 0.00**
FW(Macro-P@5)	48.00 ± 0.01	62.80 ± 0.02	58.83 ± 0.02	58.86 ± 0.05	38.32 ± 0.05	69.15 ± 0.02	40.06 ± 0.04	30.21 ± 0.04	79.60 ± 0.04
FW(Macro-R@5)	39.20 ± 0.00	64.71 ± 0.00	45.65 ± 0.00	27.03 ± 0.00	*69.02 ± 0.00*	*84.50 ± 0.00*	34.51 ± 0.00	24.01 ± 0.00	89.10 ± 0.00
FW(Macro-BA@5)	39.58 ± 0.00	65.10 ± 0.00	46.28 ± 0.00	27.06 ± 0.00	*69.02 ± 0.00*	*84.50 ± 0.00*	34.54 ± 0.00	24.04 ± 0.00	89.10 ± 0.00
FW(Macro-F_1@5)	56.90 ± 0.00	78.96 ± 0.00	66.95 ± 0.00	54.31 ± 0.00	53.89 ± 0.00	76.93 ± 0.00	*51.31 ± 0.00*	40.26 ± 0.00	83.86 ± 0.00
FW(Macro-JS@5)	56.05 ± 0.00	78.75 ± 0.00	67.09 ± 0.00	50.68 ± 0.00	55.71 ± 0.00	77.85 ± 0.00	49.82 ± 0.00	38.77 ± 0.00	84.75 ± 0.00
AmazonCat-14K									
Top-5	**54.63**	63.18	* **83.63**	41.83	36.67	68.33	36.57	27.14	69.59
PS-5	*54.61*	*64.14*	*83.57*	38.93	47.22	73.60	40.47	29.90	79.29
Pow-5$_{\beta=0.25}$	53.57	64.04	82.16	36.10	49.80	74.89	39.75	29.10	79.61
Pow-5$_{\beta=0.5}$	48.58	60.28	75.18	26.96	57.95	78.97	34.54	24.09	84.12
Log-5	54.05	**64.15**	82.61	39.68	45.08	72.53	39.95	29.60	76.51
Macro-R$_{\text{prior}}$-5	31.64	43.57	44.32	17.82	64.21	82.09	24.33	16.16	87.11
Macro-BA$_{\text{prior}}$-5	31.87	43.80	45.17	17.84	64.21	82.09	24.35	16.17	87.11
BCA(Macro-P@5)	25.91 ± 0.06	26.45 ± 0.14	47.73 ± 0.08	**65.03 ± 0.00**	8.92 ± 0.00	54.45 ± 0.00	12.89 ± 0.01	8.54 ± 0.01	70.53 ± 0.02
BCA(Macro-R@5)	29.63 ± 0.08	41.21 ± 0.07	39.83 ± 0.25	17.94 ± 0.00	**64.64 ± 0.00**	**82.31 ± 0.00**	23.87 ± 0.01	16.02 ± 0.00	*88.18 ± 0.00*
BCA(Macro-BA@5)	29.94 ± 0.07	41.52 ± 0.07	40.58 ± 0.11	17.98 ± 0.01	**64.64 ± 0.00**	**82.31 ± 0.00**	23.91 ± 0.02	16.06 ± 0.01	*88.18 ± 0.01*
BCA(Macro-F_1@5)	49.73 ± 0.00	59.48 ± 0.00	77.24 ± 0.00	47.74 ± 0.01	43.79 ± 0.00	71.89 ± 0.00	**43.76 ± 0.00**	*32.44 ± 0.00*	83.29 ± 0.01
BCA(Macro-JS@5)	49.70 ± 0.00	59.44 ± 0.00	77.20 ± 0.00	47.78 ± 0.01	43.77 ± 0.01	71.88 ± 0.00	**43.76 ± 0.00**	**32.45 ± 0.00**	83.23 ± 0.01
BCA(Cov@5)	2.90 ± 0.00	7.09 ± 0.00	3.19 ± 0.00	14.33 ± 0.01	41.88 ± 0.00	70.92 ± 0.00	9.46 ± 0.01	5.46 ± 0.00	**88.26 ± 0.01**
FW(Macro-P@5)	36.75 ± 0.01	42.09 ± 0.01	59.38 ± 0.02	*54.56 ± 0.07*	25.06 ± 0.03	62.52 ± 0.02	30.50 ± 0.04	20.94 ± 0.03	76.98 ± 0.05
FW(Macro-R@5)	31.09 ± 0.00	43.06 ± 0.00	42.96 ± 0.00	17.64 ± 0.00	*64.48 ± 0.00*	*82.23 ± 0.00*	24.04 ± 0.00	15.96 ± 0.00	87.50 ± 0.00
FW(Macro-BA@5)	31.38 ± 0.00	43.35 ± 0.00	44.09 ± 0.00	17.65 ± 0.00	*64.48 ± 0.00*	*82.23 ± 0.00*	24.06 ± 0.00	15.98 ± 0.00	87.50 ± 0.00
FW(Macro-F_1@5)	49.63 ± 0.00	57.60 ± 0.00	77.08 ± 0.00	47.53 ± 0.00	43.33 ± 0.00	71.66 ± 0.00	43.08 ± 0.00	32.05 ± 0.00	80.22 ± 0.00
FW(Macro-JS@5)	49.58 ± 0.00	57.59 ± 0.00	77.00 ± 0.00	47.61 ± 0.00	43.28 ± 0.00	71.63 ± 0.00	*43.09 ± 0.00*	32.08 ± 0.00	80.10 ± 0.00

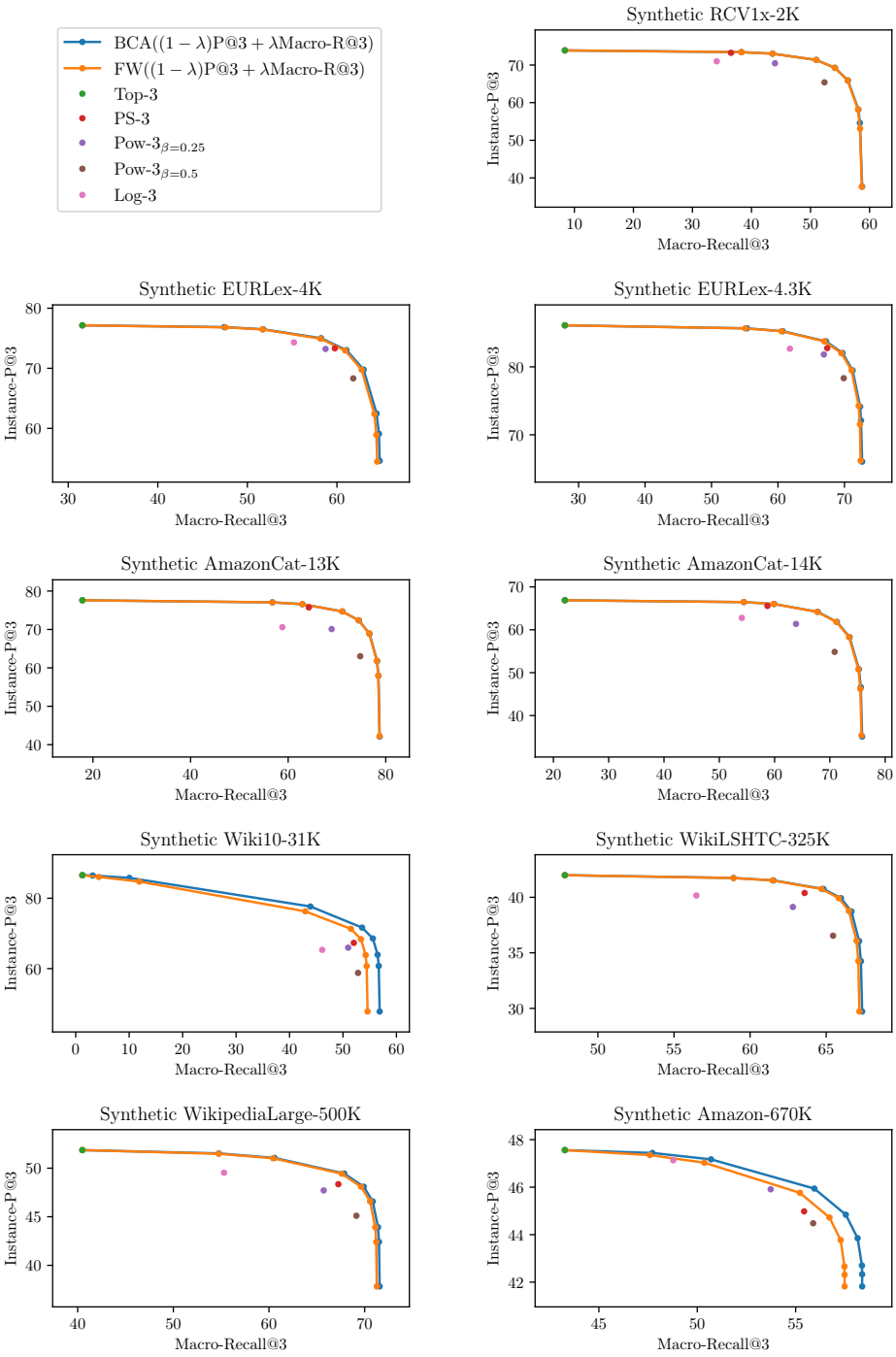
Method	Instance @5			Macro @5					
	P	PS	R	P	R	BA	F_1	JS	Cov
Wiki10-31K									
Top-5	**63.00**	97.56	* **17.98**	3.73	1.00	50.50	1.42	0.96	4.38
PS-5	*61.83*	128.33	*17.82*	7.91	3.69	51.84	4.51	3.49	9.47
Pow-$5_{\beta=0.25}$	61.06	127.25	17.56	7.40	3.39	51.69	4.19	3.20	9.01
Pow-$5_{\beta=0.5}$	54.03	146.97	15.61	10.03	5.94	52.97	6.77	5.39	12.75
Log-5	56.16	120.69	15.98	6.61	2.91	51.45	3.67	2.72	8.33
Macro-R_{prior}-5	38.04	150.76	10.94	12.04	**8.73**	**54.36**	**9.03**	*7.23*	*16.77*
Macro-BA_{prior}-5	38.28	**150.94**	11.01	12.05	*8.72*	*54.35*	**9.03**	*7.23*	16.75
BCA(Macro-P@5)	34.35 ± 0.21	102.57 ± 0.27	9.69 ± 0.06	**14.44 ± 0.01**	6.18 ± 0.02	53.08 ± 0.01	7.41 ± 0.02	6.15 ± 0.02	14.62 ± 0.01
BCA(Macro-R@5)	27.57 ± 0.00	129.69 ± 0.00	7.85 ± 0.00	12.35 ± 0.00	8.48 ± 0.00	54.23 ± 0.00	*8.98 ± 0.00*	7.21 ± 0.00	**17.43 ± 0.00**
BCA(Macro-BA@5)	27.64 ± 0.00	129.78 ± 0.00	7.87 ± 0.00	12.35 ± 0.00	8.48 ± 0.00	54.23 ± 0.00	*8.98 ± 0.00*	7.21 ± 0.00	**17.43 ± 0.00**
BCA(Macro-F_1@5)	41.01 ± 0.07	127.82 ± 0.25	11.75 ± 0.02	*13.68 ± 0.03*	6.67 ± 0.03	53.33 ± 0.01	8.08 ± 0.03	6.58 ± 0.03	15.39 ± 0.03
BCA(Macro-JS@5)	41.75 ± 0.07	128.82 ± 0.16	11.96 ± 0.03	13.50 ± 0.02	6.59 ± 0.00	53.29 ± 0.00	8.00 ± 0.00	6.50 ± 0.00	15.24 ± 0.02
BCA(Cov@5)	26.95 ± 0.05	117.21 ± 0.22	7.62 ± 0.02	11.15 ± 0.02	6.92 ± 0.02	53.45 ± 0.01	7.46 ± 0.02	5.78 ± 0.02	16.53 ± 0.03
FW(Macro-P@5)	42.77 ± 0.02	126.18 ± 0.05	12.23 ± 0.01	8.46 ± 0.00	4.82 ± 0.00	52.41 ± 0.00	5.61 ± 0.00	4.22 ± 0.00	12.36 ± 0.01
FW(Macro-R@5)	38.02 ± 0.00	150.74 ± 0.00	10.94 ± 0.00	12.04 ± 0.00	**8.73 ± 0.00**	**54.36 ± 0.00**	**9.03 ± 0.00**	**7.24 ± 0.00**	*16.77 ± 0.00*
FW(Macro-BA@5)	38.23 ± 0.00	*150.85 ± 0.00*	11.00 ± 0.00	12.05 ± 0.00	**8.73 ± 0.00**	**54.36 ± 0.00**	**9.03 ± 0.00**	**7.24 ± 0.00**	16.75 ± 0.00
FW(Macro-F_1@5)	31.55 ± 0.00	122.28 ± 0.01	8.98 ± 0.00	11.31 ± 0.00	7.74 ± 0.00	53.86 ± 0.00	7.81 ± 0.00	6.30 ± 0.00	15.25 ± 0.00
FW(Macro-JS@5)	35.10 ± 0.00	128.67 ± 0.01	10.05 ± 0.00	11.62 ± 0.00	7.77 ± 0.00	53.88 ± 0.00	7.98 ± 0.00	6.43 ± 0.00	15.41 ± 0.00
WikiLSHTC-325K									
Top-5	31.14	90.78	* 54.58	18.73	20.72	60.36	17.30	12.97	35.53
PS-5	**32.18**	101.06	**56.72**	19.62	25.45	62.72	19.69	14.64	41.44
Pow-$5_{\beta=0.25}$	*31.43*	98.92	*55.82*	19.35	24.47	62.23	19.18	14.30	40.15
Pow-$5_{\beta=0.5}$	29.97	103.63	54.41	19.12	27.76	63.88	20.28	14.90	44.18
Log-5	31.19	94.56	55.11	19.01	22.36	61.18	18.14	13.58	37.52
Macro-R_{prior}-5	27.35	**105.72**	51.42	17.50	31.21	65.60	20.16	14.33	48.56
Macro-BA_{prior}-5	27.36	**105.72**	51.42	17.50	31.21	65.60	20.16	14.33	48.56
BCA(Macro-P@5)	10.70 ± 0.01	34.79 ± 0.02	16.98 ± 0.03	**33.85 ± 0.01**	13.08 ± 0.00	56.54 ± 0.00	15.37 ± 0.00	11.82 ± 0.00	36.13 ± 0.01
BCA(Macro-R@5)	21.79 ± 0.00	96.91 ± 0.00	43.59 ± 0.00	18.23 ± 0.00	**32.81 ± 0.00**	**66.40 ± 0.00**	20.28 ± 0.00	14.43 ± 0.00	**52.70 ± 0.00**
BCA(Macro-BA@5)	21.79 ± 0.00	96.91 ± 0.00	43.59 ± 0.00	18.23 ± 0.00	**32.81 ± 0.00**	**66.40 ± 0.00**	20.28 ± 0.00	14.43 ± 0.00	**52.70 ± 0.00**
BCA(Macro-F_1@5)	24.87 ± 0.00	87.14 ± 0.01	44.61 ± 0.00	26.86 ± 0.00	26.24 ± 0.00	63.12 ± 0.00	**23.75 ± 0.00**	*18.03 ± 0.00*	48.59 ± 0.00
BCA(Macro-JS@5)	24.29 ± 0.00	82.10 ± 0.00	43.20 ± 0.01	*28.79 ± 0.00*	24.34 ± 0.00	62.17 ± 0.00	*23.69 ± 0.00*	**18.15 ± 0.00**	46.97 ± 0.00
BCA(Cov@5)	15.20 ± 0.00	78.05 ± 0.01	28.45 ± 0.01	19.08 ± 0.00	29.12 ± 0.00	64.56 ± 0.00	17.97 ± 0.00	12.72 ± 0.00	*50.61 ± 0.00*
FW(Macro-P@5)	23.91 ± 0.00	78.62 ± 0.10	43.47 ± 0.00	24.18 ± 0.00	22.06 ± 0.00	61.03 ± 0.00	20.09 ± 0.00	15.32 ± 0.00	40.17 ± 0.00
FW(Macro-R@5)	26.75 ± 0.00	*105.58 ± 0.00*	50.52 ± 0.00	17.61 ± 0.00	*31.88 ± 0.00*	*65.94 ± 0.00*	20.38 ± 0.00	14.42 ± 0.00	49.60 ± 0.00
FW(Macro-BA@5)	26.75 ± 0.00	*105.58 ± 0.00*	50.52 ± 0.00	17.61 ± 0.00	*31.88 ± 0.00*	*65.94 ± 0.00*	20.38 ± 0.00	14.42 ± 0.00	49.60 ± 0.00
FW(Macro-F_1@5)	27.13 ± 0.00	92.61 ± 0.00	48.82 ± 0.00	22.14 ± 0.00	25.88 ± 0.00	62.94 ± 0.00	21.21 ± 0.00	15.87 ± 0.00	44.13 ± 0.00
FW(Macro-JS@5)	27.28 ± 0.00	93.04 ± 0.00	48.93 ± 0.00	21.82 ± 0.00	26.03 ± 0.00	63.02 ± 0.00	21.05 ± 0.00	15.69 ± 0.00	44.32 ± 0.00

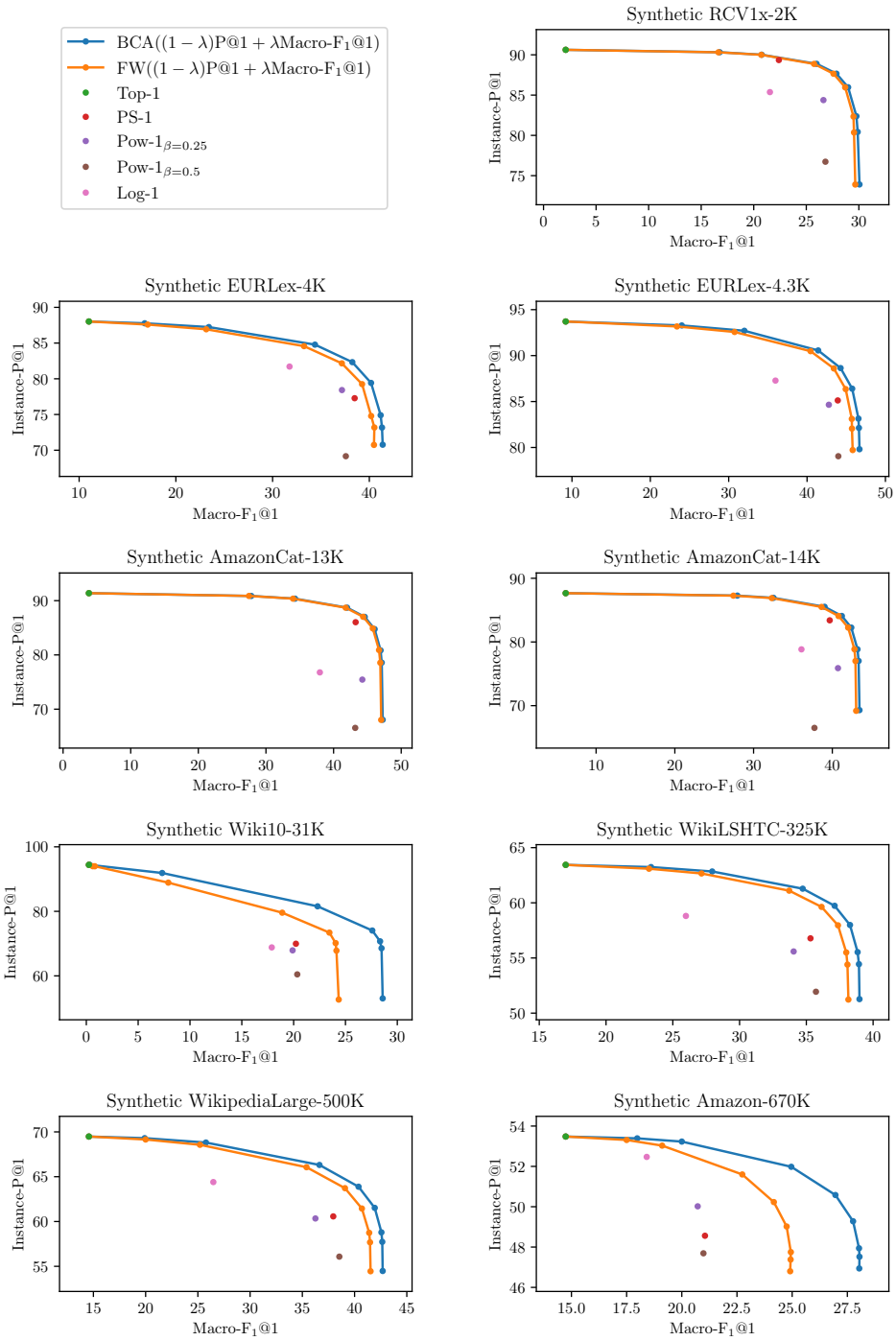
Method	Instance @5			Macro @5					
	P	PS	R	P	R	BA	F_1	JS	Cov
WikipediaLarge-500K									
Top-5	37.41	113.72	* 48.07	21.04	21.83	60.92	18.99	14.51	37.87
PS-5	**38.04**	127.08	**49.50**	23.18	27.08	63.54	22.27	16.91	44.82
Pow-$5_{\beta=0.25}$	*37.51*	124.47	*48.93*	22.72	25.88	62.94	21.59	16.42	43.28
Pow-$5_{\beta=0.5}$	35.89	130.97	47.83	23.31	29.38	64.69	**23.31**	**17.55**	47.76
Log-5	**37.43**	118.72	**48.47**	21.80	23.53	61.77	20.12	15.35	40.18
Macro-R_{prior}-5	31.75	*132.99*	43.96	22.48	33.03	66.51	**23.95**	**17.60**	52.56
Macro-BA_{prior}-5	31.75	*132.99*	43.96	22.48	**33.03**	**66.51**	**23.95**	**17.60**	52.56
BCA(Macro-P@5)	13.66 ± 0.01	46.06 ± 0.02	15.67 ± 0.02	**36.12 ± 0.00**	14.28 ± 0.00	57.14 ± 0.00	16.79 ± 0.00	13.11 ± 0.00	38.29 ± 0.00
BCA(Macro-R@5)	24.92 ± 0.00	121.88 ± 0.00	36.36 ± 0.00	23.33 ± 0.00	*34.71 ± 0.00*	**67.36 ± 0.00**	24.28 ± 0.00	17.71 ± 0.00	**57.18 ± 0.00**
BCA(Macro-BA@5)	24.92 ± 0.00	121.88 ± 0.00	36.37 ± 0.00	23.33 ± 0.00	**34.72 ± 0.00**	**67.36 ± 0.00**	24.28 ± 0.00	17.71 ± 0.00	**57.18 ± 0.00**
BCA(Macro-F_1@5)	29.35 ± 0.00	110.76 ± 0.00	38.08 ± 0.00	29.23 ± 0.00	27.87 ± 0.00	63.93 ± 0.00	**25.62 ± 0.00**	*19.60 ± 0.00*	51.60 ± 0.00
BCA(Macro-JS@5)	28.80 ± 0.00	105.36 ± 0.00	36.87 ± 0.00	*30.82 ± 0.00*	26.21 ± 0.00	63.10 ± 0.00	*25.46 ± 0.00*	**19.64 ± 0.00**	50.12 ± 0.00
BCA(Cov@5)	17.82 ± 0.00	98.49 ± 0.01	25.15 ± 0.00	24.08 ± 0.01	30.93 ± 0.00	65.46 ± 0.00	21.85 ± 0.00	15.87 ± 0.00	*55.71 ± 0.00*
FW(Macro-P@5)	24.71 ± 0.00	88.82 ± 0.00	32.65 ± 0.00	26.36 ± 0.00	21.40 ± 0.00	60.70 ± 0.00	20.20 ± 0.00	15.66 ± 0.00	40.13 ± 0.00
FW(Macro-R@5)	31.02 ± 0.00	**133.28 ± 0.00**	43.14 ± 0.00	22.59 ± 0.00	33.91 ± 0.00	*66.95 ± 0.00*	24.15 ± 0.00	17.64 ± 0.00	53.83 ± 0.00
FW(Macro-BA@5)	31.02 ± 0.00	**133.28 ± 0.00**	43.14 ± 0.00	22.59 ± 0.00	33.91 ± 0.00	*66.95 ± 0.00*	24.15 ± 0.00	17.64 ± 0.00	53.83 ± 0.00
FW(Macro-F_1@5)	30.59 ± 0.00	114.01 ± 0.00	40.12 ± 0.00	25.14 ± 0.00	27.22 ± 0.00	63.61 ± 0.00	22.98 ± 0.00	17.43 ± 0.00	47.21 ± 0.00
FW(Macro-JS@5)	30.87 ± 0.00	110.45 ± 0.00	39.45 ± 0.00	25.12 ± 0.00	25.84 ± 0.00	62.92 ± 0.00	22.19 ± 0.00	16.89 ± 0.00	44.90 ± 0.00
Amazon-670K									
Top-5	36.71	259.09	* 34.39	14.21	14.41	57.20	13.39	11.41	19.79
PS-5	36.71	272.96	34.51	15.24	15.68	57.84	14.50	12.35	21.41
Pow-$5_{\beta=0.25}$	**36.89**	270.81	**34.65**	15.10	15.43	57.71	14.32	12.22	21.07
Pow-$5_{\beta=0.5}$	36.49	**275.49**	34.39	15.58	16.01	58.00	14.81	12.61	21.81
Log-5	*36.84*	264.86	*34.55*	14.61	14.87	57.43	13.81	11.78	20.36
Macro-R_{prior}-5	34.70	*273.87*	32.86	16.02	16.39	*58.19*	15.06	12.76	22.30
Macro-BA_{prior}-5	34.70	*273.87*	32.86	16.02	16.39	*58.19*	15.06	12.76	22.30
BCA(Macro-P@5)	24.08 ± 0.00	200.71 ± 0.08	23.12 ± 0.01	**21.04 ± 0.01**	13.67 ± 0.01	56.84 ± 0.00	15.33 ± 0.01	13.52 ± 0.01	21.38 ± 0.01
BCA(Macro-R@5)	33.28 ± 0.00	265.95 ± 0.00	31.56 ± 0.00	17.62 ± 0.00	*16.44 ± 0.00*	**58.22 ± 0.00**	15.84 ± 0.00	*13.57 ± 0.00*	**22.98 ± 0.00**
BCA(Macro-BA@5)	33.28 ± 0.00	265.95 ± 0.00	31.56 ± 0.00	17.62 ± 0.00	*16.44 ± 0.00*	**58.22 ± 0.00**	15.84 ± 0.00	*13.57 ± 0.00*	**22.98 ± 0.00**
BCA(Macro-F_1@5)	32.25 ± 0.01	250.48 ± 0.05	30.52 ± 0.01	20.00 ± 0.01	15.47 ± 0.01	57.74 ± 0.00	**16.47 ± 0.01**	**14.44 ± 0.01**	22.61 ± 0.01
BCA(Macro-JS@5)	31.91 ± 0.00	247.25 ± 0.02	30.19 ± 0.00	*20.14 ± 0.01*	15.27 ± 0.00	57.64 ± 0.00	*16.40 ± 0.00*	**14.44 ± 0.00**	22.34 ± 0.01
BCA(Cov@5)	29.91 ± 0.01	248.56 ± 0.03	28.10 ± 0.01	16.50 ± 0.00	15.63 ± 0.00	57.81 ± 0.00	14.83 ± 0.00	12.49 ± 0.00	*22.78 ± 0.00*
FW(Macro-P@5)	33.75 ± 0.01	256.96 ± 0.06	31.85 ± 0.01	15.80 ± 0.01	15.05 ± 0.00	57.52 ± 0.00	14.24 ± 0.00	12.12 ± 0.00	20.73 ± 0.01
FW(Macro-R@5)	34.31 ± 0.00	273.24 ± 0.00	32.53 ± 0.00	16.15 ± 0.00	**16.45 ± 0.00**	**58.22 ± 0.00**	15.10 ± 0.00	12.79 ± 0.00	22.36 ± 0.00
FW(Macro-BA@5)	34.31 ± 0.00	273.24 ± 0.00	32.53 ± 0.00	16.15 ± 0.00	**16.45 ± 0.00**	**58.22 ± 0.00**	15.10 ± 0.00	12.79 ± 0.00	22.36 ± 0.00
FW(Macro-F_1@5)	34.91 ± 0.00	262.95 ± 0.00	32.86 ± 0.00	15.60 ± 0.00	15.25 ± 0.00	57.62 ± 0.00	14.32 ± 0.00	12.19 ± 0.00	20.94 ± 0.00
FW(Macro-JS@5)	34.49 ± 0.00	259.77 ± 0.00	32.47 ± 0.00	15.59 ± 0.00	15.07 ± 0.00	57.54 ± 0.00	14.21 ± 0.00	12.11 ± 0.00	20.73 ± 0.00

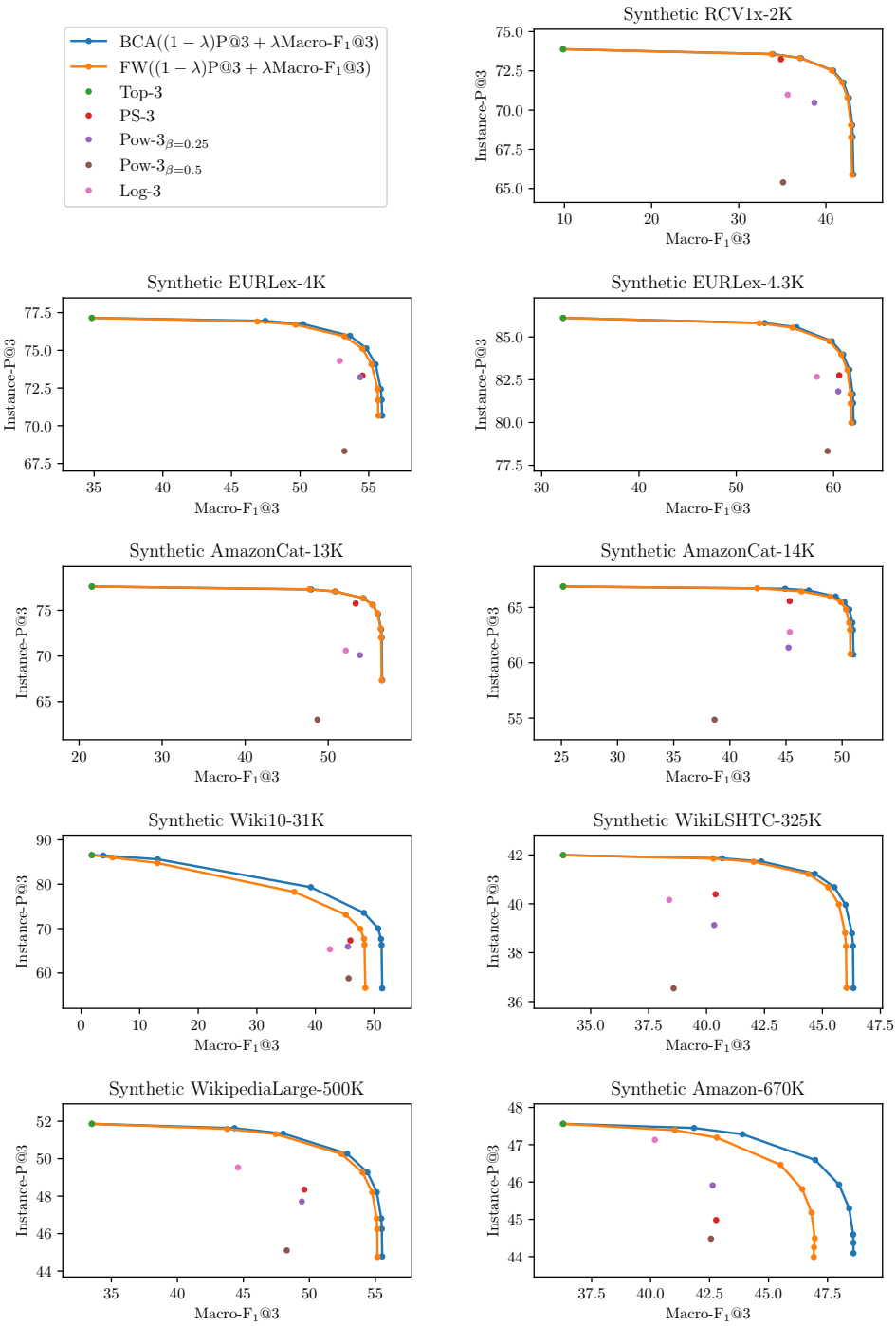
B.2.3 Optimization of mixed utilities on datasets with synthetic labels

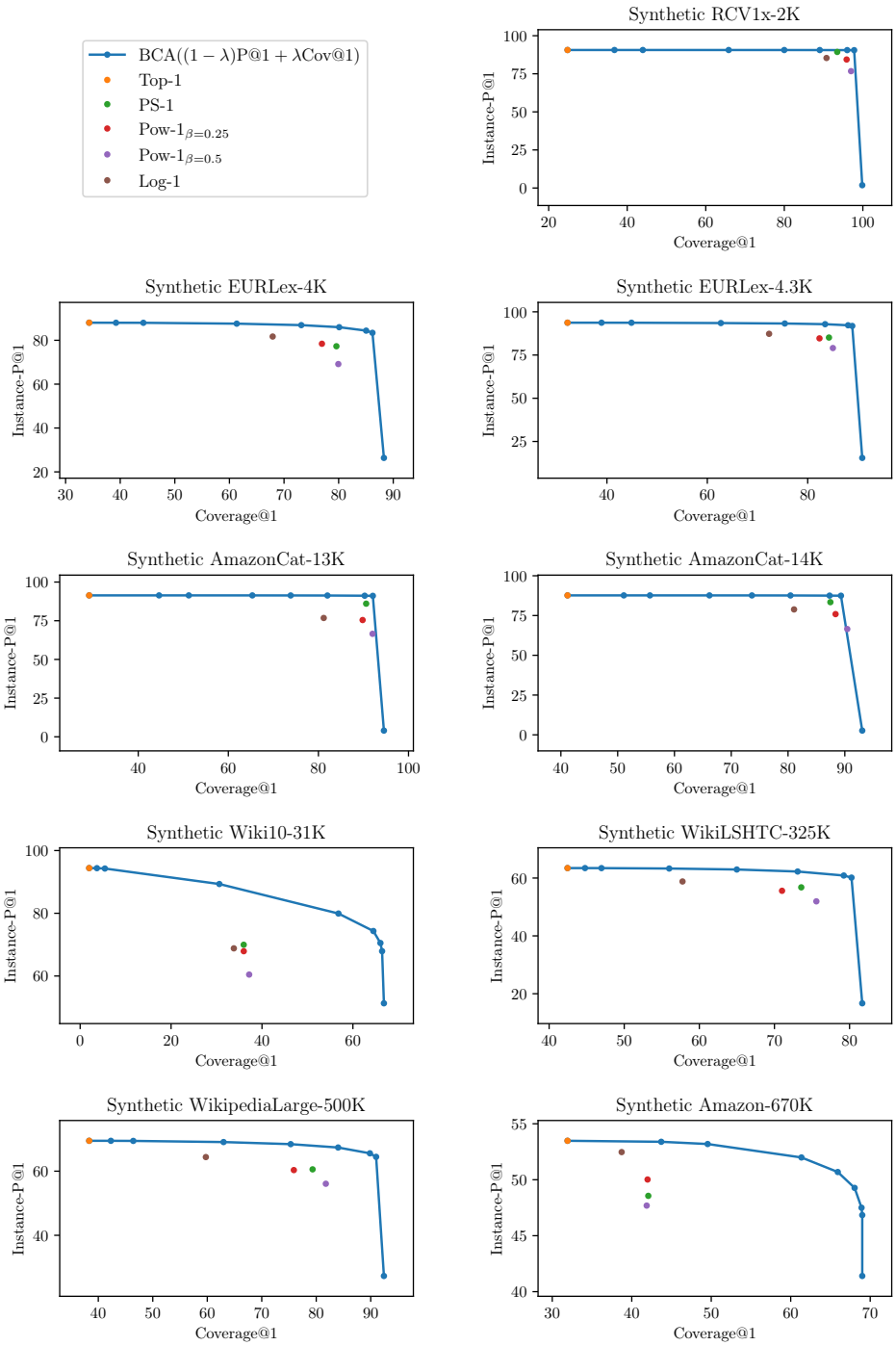
Figure B.1: Results (%) for $k \in \{1, 3\}$ of optimization of mixed utilities on synthetic versions of XMLC datasets with ideal estimates of marginal conditional probabilities $\boldsymbol{\eta}(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}(\boldsymbol{x})$.

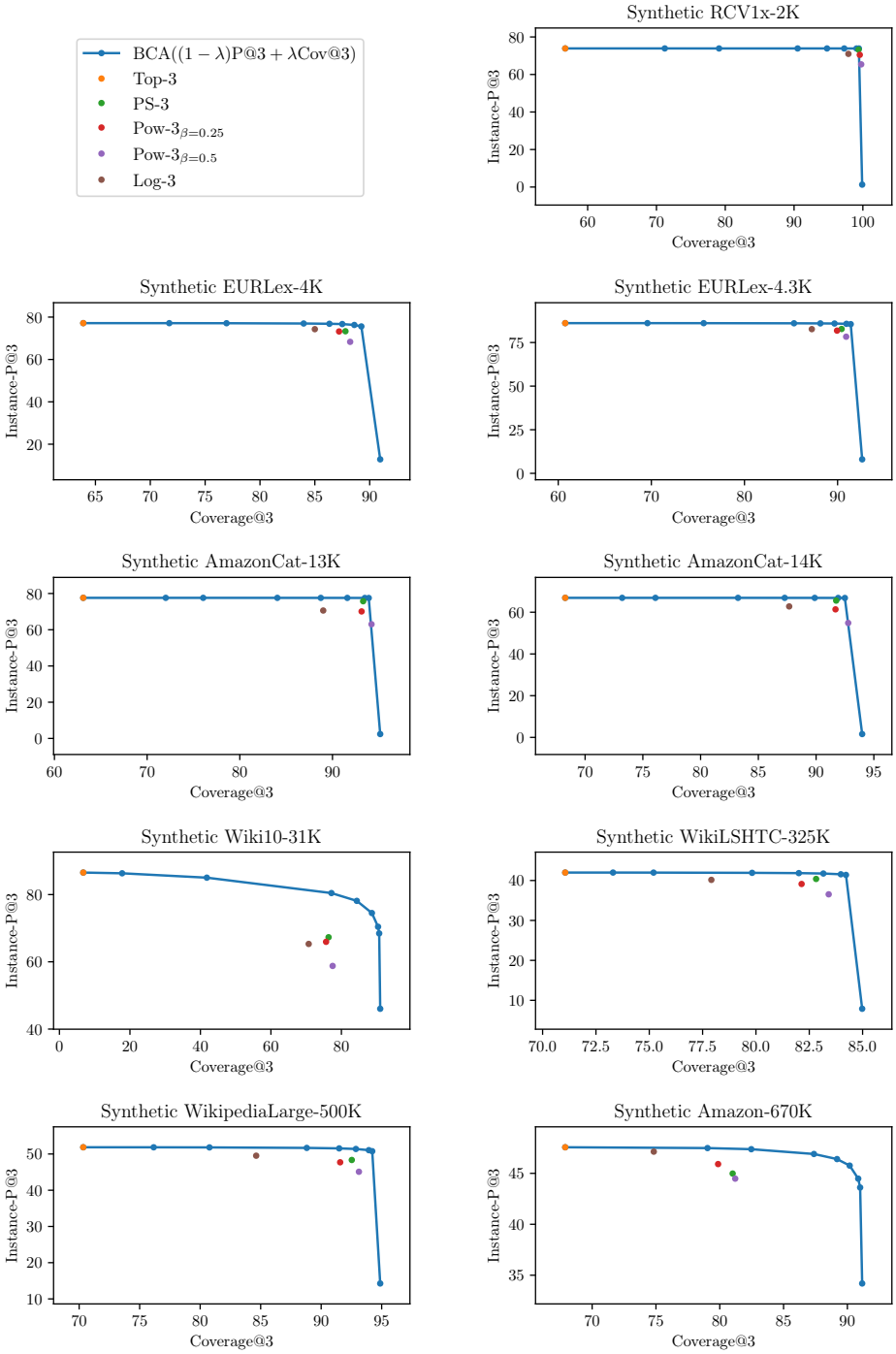



BCA((1 − λ)P@3 + λMacro-R@3)
FW((1 − λ)P@3 + λMacro-R@3)
Top-3
PS-3
Pow-3β=0.25
Pow-3β=0.5
Log-3
Synthetic RCV1x-2K
Synthetic EURLex-4K
Synthetic EURLex-4.3K
Synthetic AmazonCat-13K
Synthetic AmazonCat-14K
Synthetic Wiki10-31K
Synthetic WikiLSHTC-325K
Synthetic WikipediaLarge-500K
Synthetic Amazon-670K
Instance-P@3
Macro-Recall@3


BCA($(1-\lambda)$P@1 + λMacro-F$_1$@1)
FW($(1-\lambda)$P@1 + λMacro-F$_1$@1)
Top-1
PS-1
Pow-1$_{\beta=0.25}$
Pow-1$_{\beta=0.5}$
Log-1
Synthetic RCV1x-2K
Synthetic EURLex-4K
Synthetic EURLex-4.3K
Synthetic AmazonCat-13K
Synthetic AmazonCat-14K
Synthetic Wiki10-31K
Synthetic WikiLSHTC-325K
Synthetic WikipediaLarge-500K
Synthetic Amazon-670K
Instance-P@1
Macro-F$_1$@1

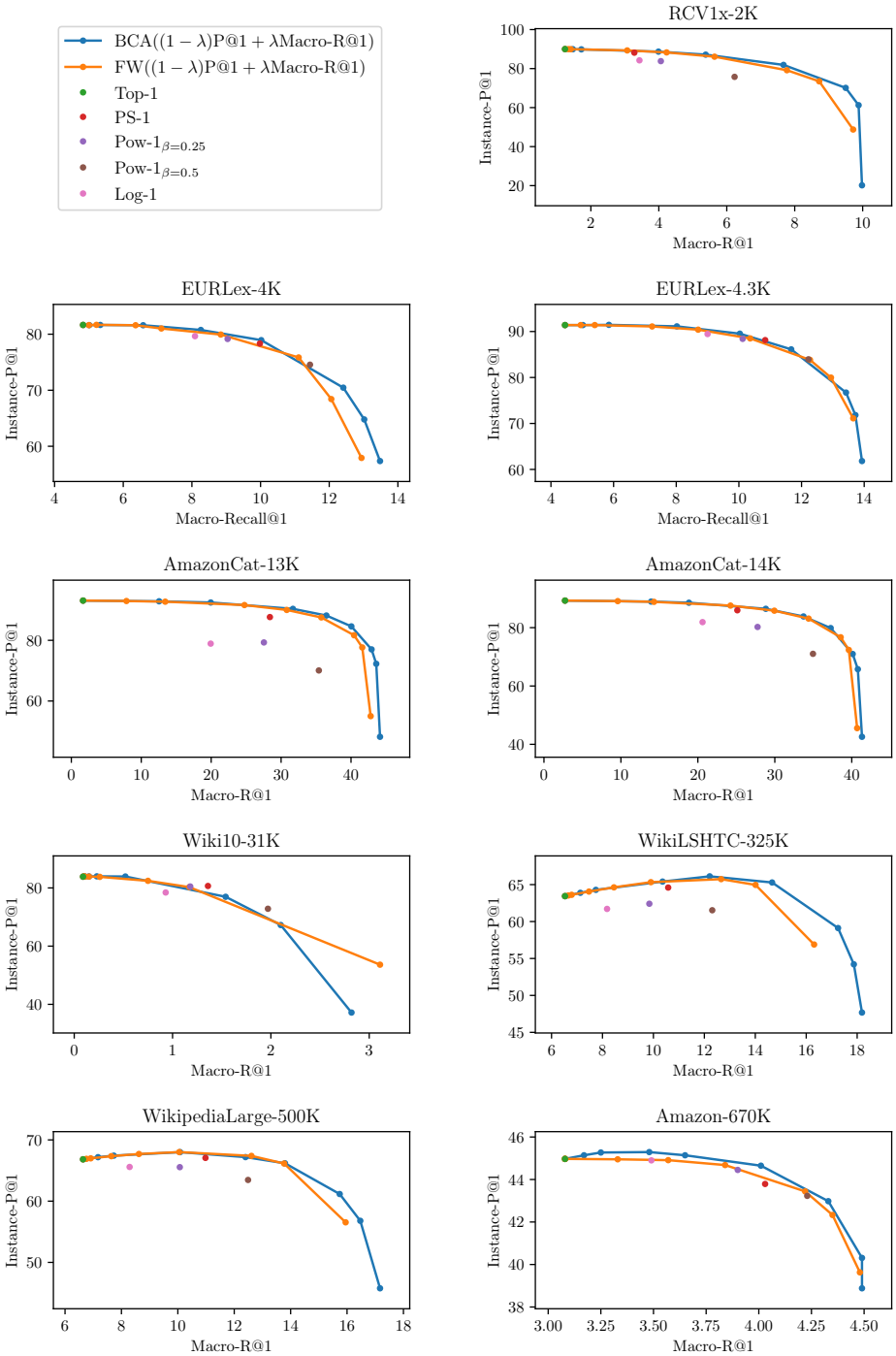

BCA((1 − λ)P@3 + λMacro-F1@3)
FW((1 − λ)P@3 + λMacro-F1@3)
Top-3
PS-3
Pow-3β=0.25
Pow-3β=0.5
Log-3
Synthetic RCV1x-2K
Synthetic EURLex-4K
Synthetic EURLex-4.3K
Synthetic AmazonCat-13K
Synthetic AmazonCat-14K
Synthetic Wiki10-31K
Synthetic WikiLSHTC-325K
Synthetic WikipediaLarge-500K
Synthetic Amazon-670K
Instance-P@3
Macro-F1@3

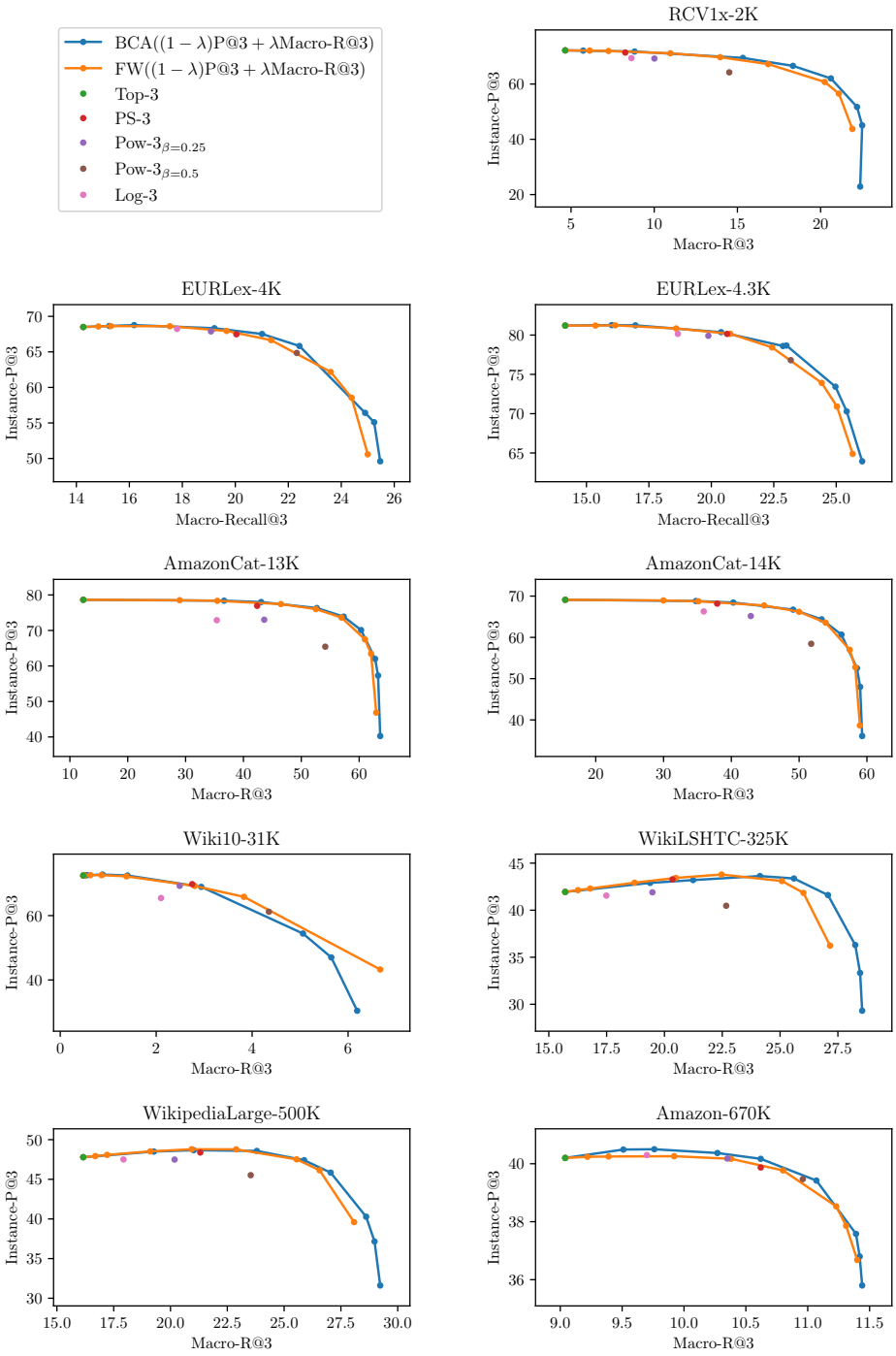

BCA((1 − λ)P@1 + λCov@1)
Top-1
PS-1
Pow-1β=0.25
Pow-1β=0.5
Log-1
Synthetic RCV1x-2K
Synthetic EURLex-4K
Synthetic EURLex-4.3K
Synthetic AmazonCat-13K
Synthetic AmazonCat-14K
Synthetic Wiki10-31K
Synthetic WikiLSHTC-325K
Synthetic WikipediaLarge-500K
Synthetic Amazon-670K
Instance-P@1
Coverage@1

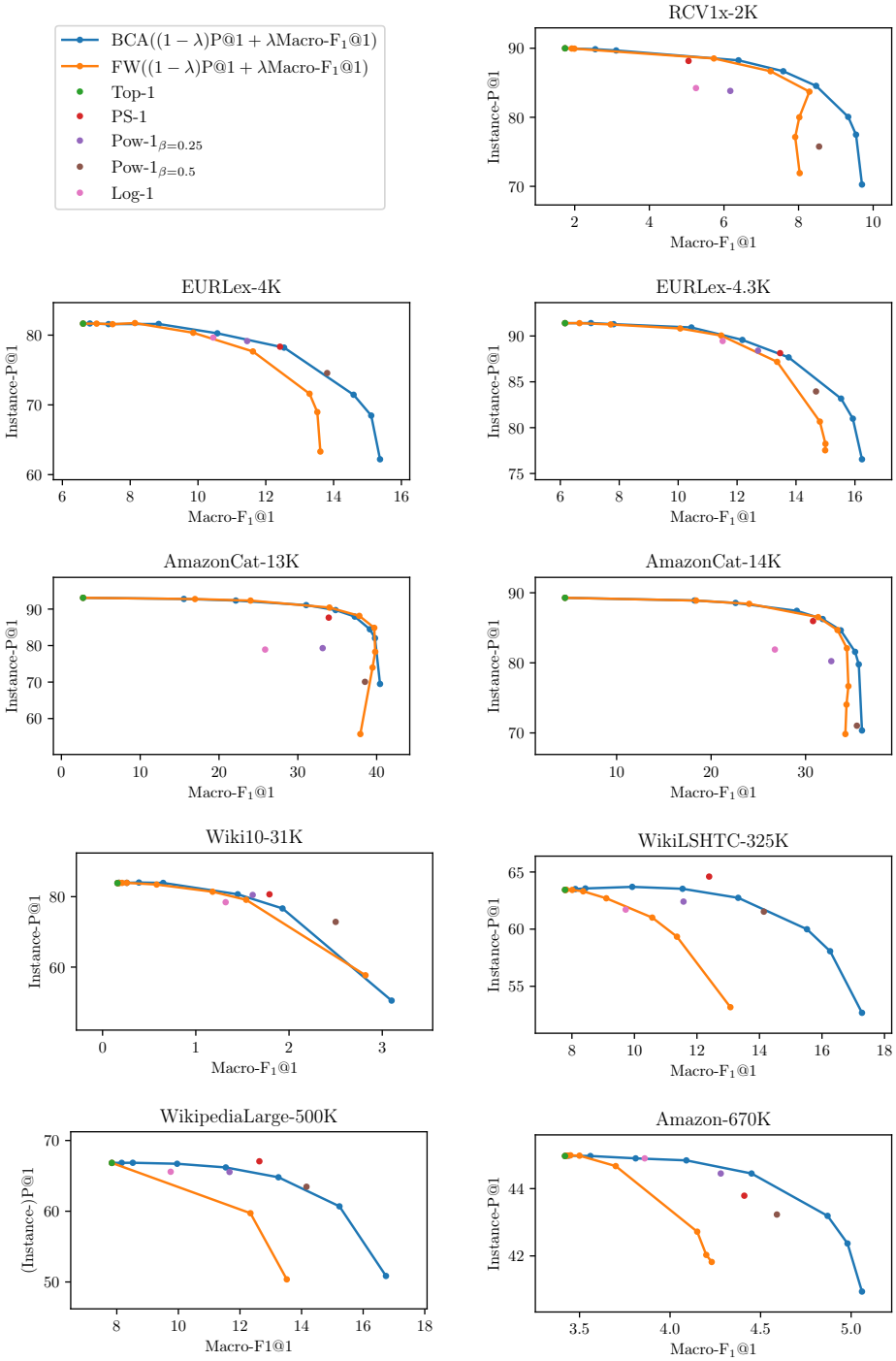

BCA((1 − λ)P@3 + λCov@3)
Top-3
PS-3
Pow-3β=0.25
Pow-3β=0.5
Log-3
Synthetic RCV1x-2K
Synthetic EURLex-4K
Synthetic EURLex-4.3K
Synthetic AmazonCat-13K
Synthetic AmazonCat-14K
Synthetic Wiki10-31K
Synthetic WikiLSHTC-325K
Synthetic WikipediaLarge-500K
Synthetic Amazon-670K
Instance-P@3
Coverage@3

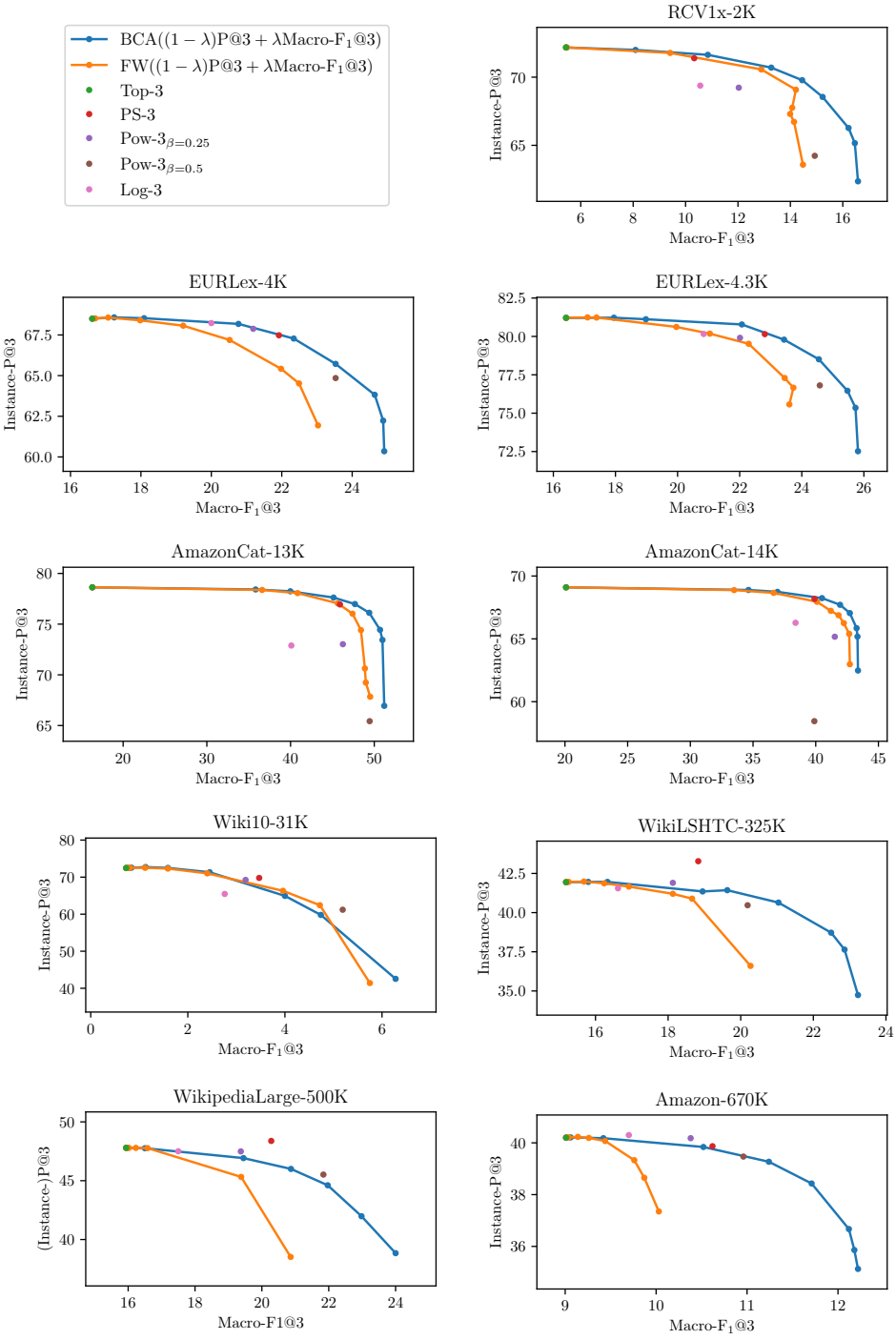
B.2.4 Optimization of mixed utilities on original datasets

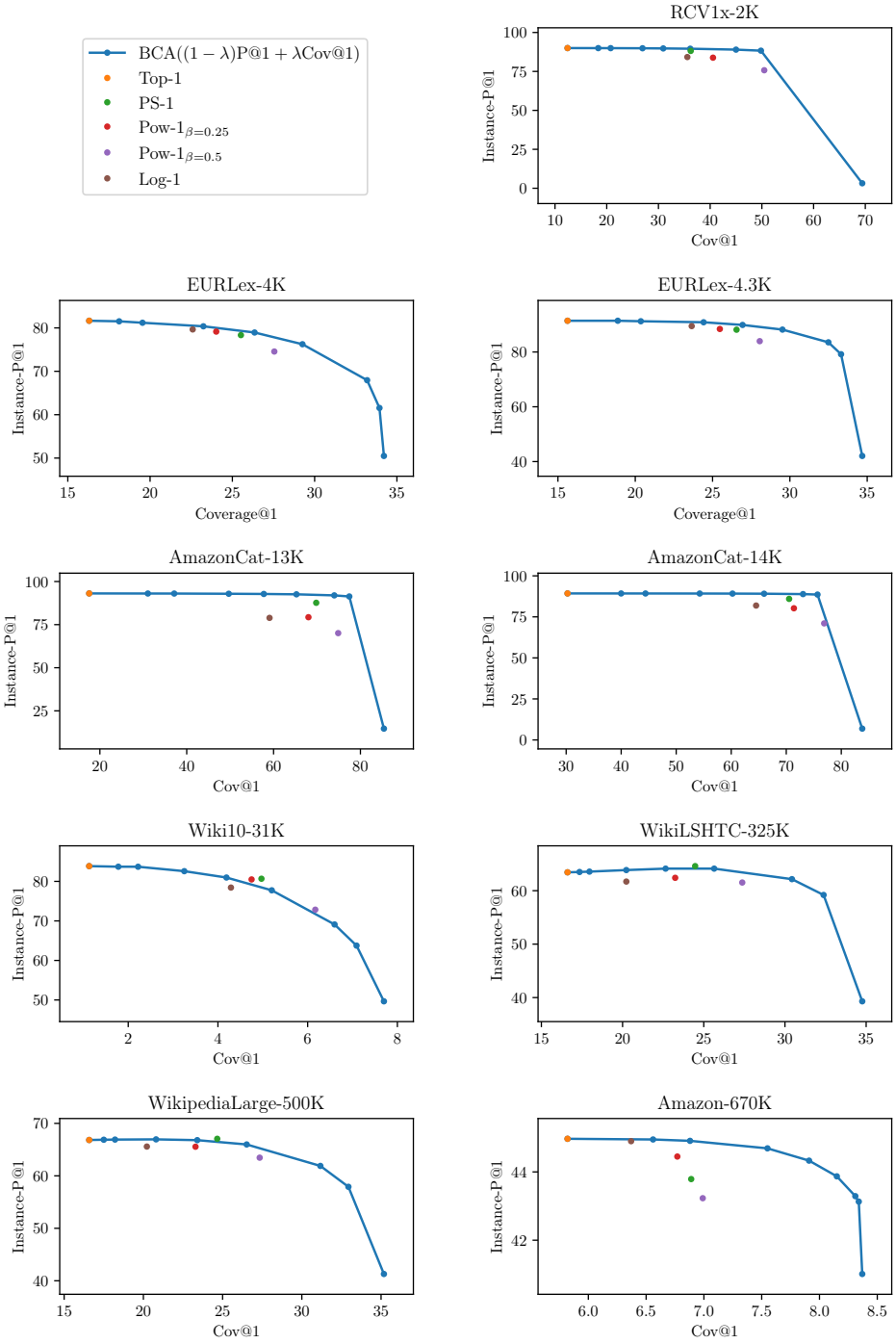
Figure B.2: Results (%) for $k \in \{1, 3\}$ of optimization of mixed utilities on original XMLC datasets with marginal conditional probabilities coming from the PLT model.

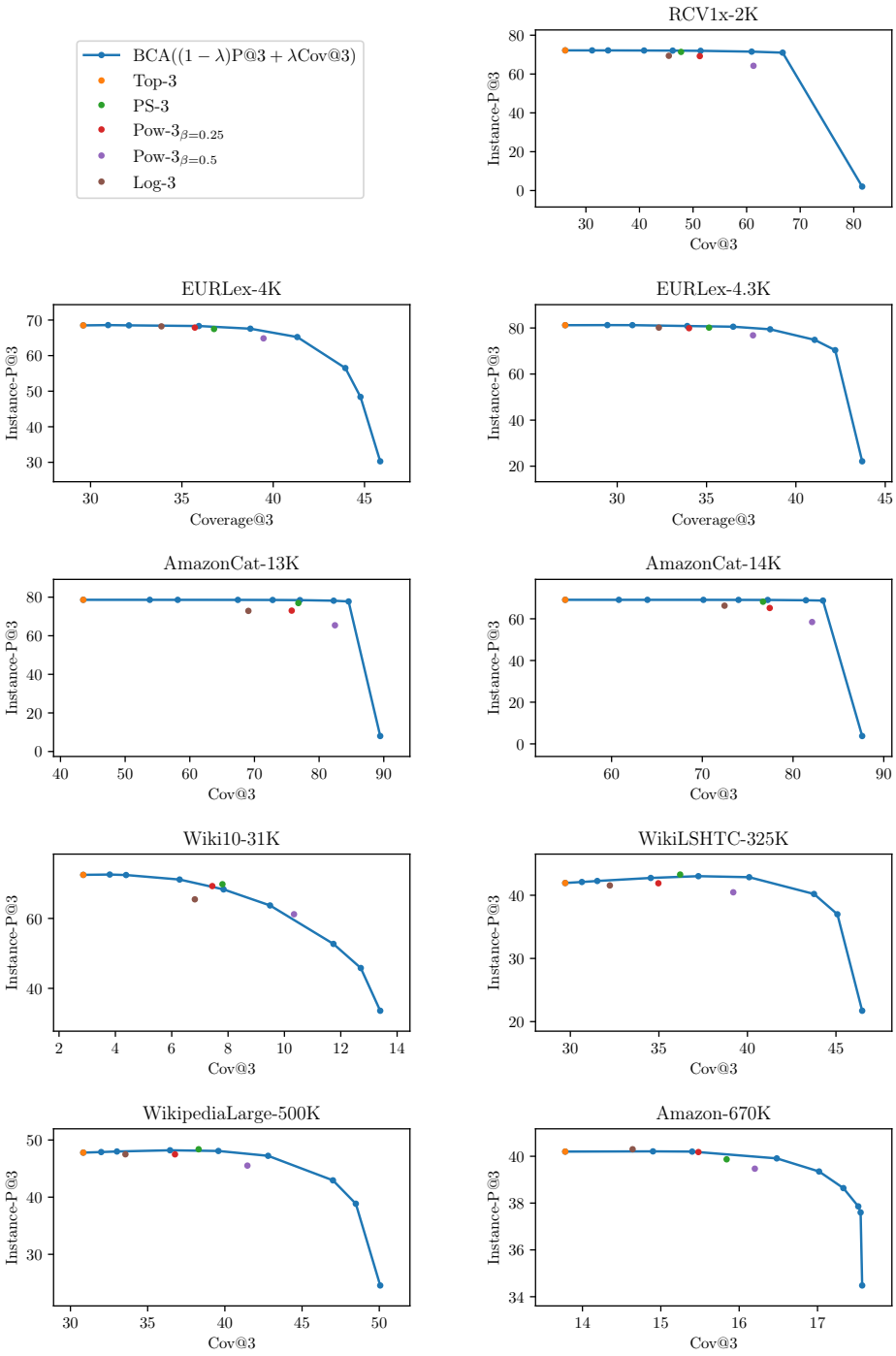



BCA((1 − λ)P@3 + λMacro-R@3)
FW((1 − λ)P@3 + λMacro-R@3)
Top-3
PS-3
Pow-3β=0.25
Pow-3β=0.5
Log-3
RCV1x-2K
EURLex-4K
EURLex-4.3K
AmazonCat-13K
AmazonCat-14K
Wiki10-31K
WikiLSHTC-325K
WikipediaLarge-500K
Amazon-670K
Instance-P@3
Macro-R@3
Macro-Recall@3


BCA((1 − λ)P@1 + λMacro-F1@1)
FW((1 − λ)P@1 + λMacro-F1@1)
Top-1
PS-1
Pow-1β=0.25
Pow-1β=0.5
Log-1
RCV1x-2K
EURLex-4K
EURLex-4.3K
AmazonCat-13K
AmazonCat-14K
Wiki10-31K
WikiLSHTC-325K
WikipediaLarge-500K
Amazon-670K
Instance-P@1
(Instance-)P@1
Macro-F1@1
Macro-F1@1


BCA((1 − λ)P@3 + λMacro-F1@3)
FW((1 − λ)P@3 + λMacro-F1@3)
Top-3
PS-3
Pow-3β=0.25
Pow-3β=0.5
Log-3
RCV1x-2K
Instance-P@3
Macro-F1@3
EURLex-4K
EURLex-4.3K
AmazonCat-13K
AmazonCat-14K
Wiki10-31K
WikiLSHTC-325K
WikipediaLarge-500K
(Instance-)P@3
Macro-F1@3
Amazon-670K


BCA((1 − λ)P@1 + λCov@1)
Top-1
PS-1
Pow-1β=0.25
Pow-1β=0.5
Log-1
RCV1x-2K
EURLex-4K
EURLex-4.3K
AmazonCat-13K
AmazonCat-14K
Wiki10-31K
WikiLSHTC-325K
WikipediaLarge-500K
Amazon-670K
Instance-P@1
Coverage@1
Cov@1


BCA((1 − λ)P@3 + λCov@3)
Top-3
PS-3
Pow-3β=0.25
Pow-3β=0.5
Log-3
RCV1x-2K
Instance-P@3
Cov@3
EURLex-4K
Coverage@3
EURLex-4.3K
AmazonCat-13K
AmazonCat-14K
Wiki10-31K
WikiLSHTC-325K
WikipediaLarge-500K
Amazon-670K