GENERALIZED TEST UTILITIES FOR LONG-TAIL PERFORMANCE IN EXTREME MULTI-LABEL CLASSIFICATION

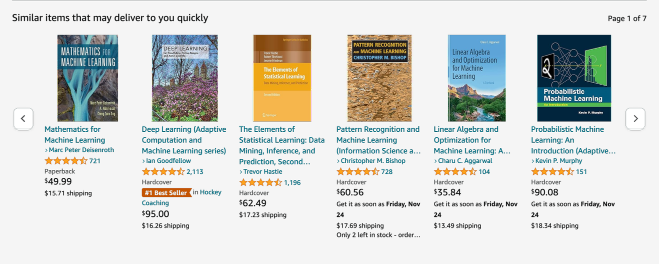
Erik Schultheis[1] Marek Wydmuch[2] Wojtek Kotłowski[2] Rohit Babbar[1,3] Krzysztof Dembczyński[2,4]

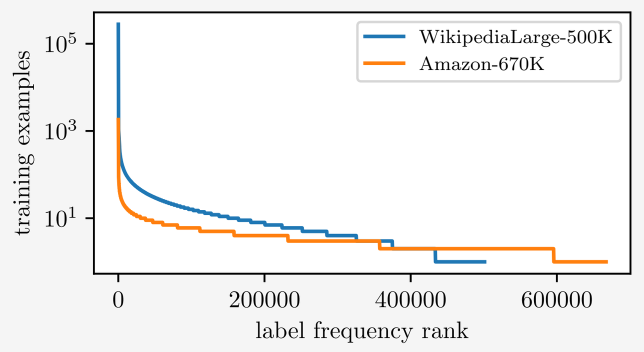
[1]Aalto University, Helsinki, Finland [2]Poznan University of Technology, Poland [3]University of Bath, UK [4]Yahoo Research, New York, USA

Extreme Multi-Label Classification (XMLC)

- Multi-label classification: $\boldsymbol{x} \in \mathcal{X} \to \boldsymbol{y} \in \mathcal{Y} := \{0,1\}^m$,
- with **extreme** number of labels ($m \geq 10^5$),
- and label vector $\boldsymbol{y}$ being **very sparse** ($\|\boldsymbol{y}\|_1 \ll m$).

- Many problems are naturally **budgeted at k**, i.e., one needs to make exactly k predictions ($\|\hat{\boldsymbol{y}}\|_1 = k$).
- The labels follow a **long-tail** distribution.
- The rare labels are considered more **"rewarding"**.





The problem with common performance metrics

- Commonly used metrics (e.g., Precision@k, Propensity-Scored Precision@k) are **insensitive** to tail labels performance.
- There is a need to use **new metrics** for that purpose.
- Experiment: evaluating a classifier trained with 1000 most popular labels, but tested on the full set (AmazonCat-13k):

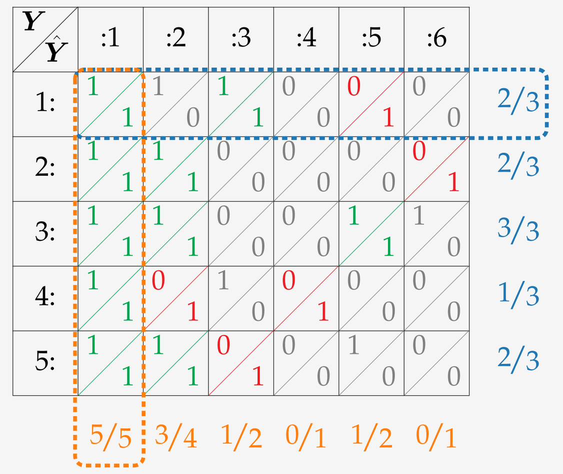
Metric	full labels @1	full labels @3	full labels @5	head labels @1 (diff.)	head labels @3 (diff.)	head labels @5 (diff.)
Precision	93	79	64	93 (+0.1%)	76 (-2.7%)	58 (-8.7%)
PS-Precision	50	63	70	49 (-1.4%)	58 (-7.8%)	57 (-18%)
Macro-Precision	13	33	44	4.3 (-68%)	5.3 (-84%)	4.3 (-90%)
Macro-Recall	1.4	11	31	0.5 (-66%)	2.7 (-76%)	4.1 (-87%)
Macro-F1	2.3	15	33	0.7 (-67%)	3.1 (-79%)	3.8 (-89%)
Coverage	15	41	61	5.1 (-66%)	7.4 (-82%)	7.5 (-88%)

Macro-averaged performance measures are sensitive to tail labels performance!

Generalized performance measures at k

Instead of calculating and averaging performance **instance-wise**, calculate per-label ψ^j and average over labels (**label-wise**):

$$\Psi@k(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) = \sum_{j=1}^{m} \psi^j(\boldsymbol{y}_{:j}, \hat{\boldsymbol{y}}_{:j}) \,.$$



Precision@3: instance ≈ 66%, macro ≈ 46%

- Can balance contribution from all labels.
- Reduces to macro-average if ψ^j are equal.
- Covers also (instance) Precision@k and PS-Precision@k.

Metric	$\psi(\text{tp}, \text{fp}, \text{fn}, \text{tn})$	$\psi(t, q, p)$
Precision	$\frac{\text{tp}}{\text{tp}+\text{fp}}$	$\frac{t}{q}$
Recall	$\frac{\text{tp}}{\text{tp}+\text{fn}}$	$\frac{t}{p}$
F_1	$\frac{2\text{tp}}{2\text{tp}+\text{fn}+\text{fp}}$	$\frac{2t}{p+q}$
Coverage	$\mathbb{1}[\text{tp} > 0]$	$\mathbb{1}[t > 0]$
Jaccard, G-Mean, etc.		

Generalized performance measures can be written as a function of **label-wise confusion matrix**:

$$\boldsymbol{C}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) := [\boldsymbol{C}(\boldsymbol{y}_{:1}, \hat{\boldsymbol{y}}_{:1}), \ldots, \boldsymbol{C}(\boldsymbol{y}_{:m}, \hat{\boldsymbol{y}}_{:m})] \,.$$

This matrix is a vector of **binary confusion matrices** for each label:

$$\boldsymbol{C}(\boldsymbol{y}, \hat{\boldsymbol{y}}) := \begin{pmatrix} \text{tn} := \frac{1}{n}\sum_{i=1}^{n}(1-y_i)(1-\hat{y}_i) & \text{fp} := \frac{1}{n}\sum_{i=1}^{n}(1-y_i)\hat{y}_i \\ \text{fn} := \frac{1}{n}\sum_{i=1}^{n} y_i(1-\hat{y}_i) & \text{tp} := \frac{1}{n}\sum_{i=1}^{n} y_i\hat{y}_i \end{pmatrix} .$$

Reparameterization: $t = \text{tp}$, $q = \text{tp} + \text{fp}$, $p = \text{tp} + \text{fn}$.

true positive rate | predicted positive rate | positive rate

Expected Test-Utility Framework

Two **conflicting** statistical frameworks:

Expected Test-Utiltiy (ETU): Given a **specific set of instances** with unknown, stochastic labels, how well can we expect a classifier to perform:

$$\Psi_{ETU} = \mathbb{E}_{\boldsymbol{Y}|\boldsymbol{X}}[\Psi(\mathbf{C}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}))]$$

Population Utility (PU): Given a **distribution of instances**, how well can we expect a classifier to perform on the population level, i.e., for an infinite sample:

$$\Psi_{PU} = \Psi\left(\mathbb{E}_{\boldsymbol{y}}[\mathbf{C}(\boldsymbol{y}, \hat{\boldsymbol{y}})]\right) .$$

The ETU framework is relevant, e.g., if recommendations for the entire catalog of products are re-generated on a daily basis.

Optimal predictions under ETU

Instance conditioned **marginal probability** of a label: $\eta_j(\boldsymbol{x}) := \mathbb{P}[y_j = 1 \mid \boldsymbol{x}]$.

$$\mathbb{E}_{\boldsymbol{Y}|\boldsymbol{X}}[\Psi(\mathbf{C}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}))] = \sum_{j=1}^{m} \sum_{\boldsymbol{y}' \in \{0,1\}^n} \left(\prod_{i=1}^{n} \eta_j(\boldsymbol{x}_i) y'_i + (1-\eta_j(\boldsymbol{x}_i))(1-y'_i) \right) \psi^j(\boldsymbol{C}(\boldsymbol{y}', \hat{\boldsymbol{y}}_{:j})) \,.$$

requires summing over 2^n summands $\boldsymbol{y}'$ | the marginals are enough

Semi-empirical ETU approximation

A **computationally easier approach** by taking the expectation over the labels:

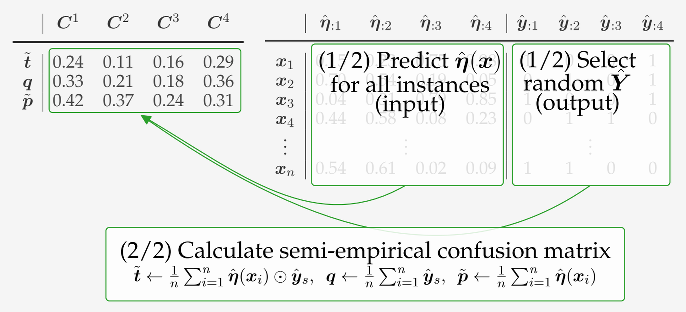
$$\tilde{\Psi}_{\text{ETU}} := \Psi(\mathbb{E}_{\boldsymbol{Y}|\boldsymbol{X}}[\boldsymbol{t}], \mathbb{E}_{\boldsymbol{Y}|\boldsymbol{X}}[\boldsymbol{q}], \mathbb{E}_{\boldsymbol{Y}|\boldsymbol{X}}[\boldsymbol{p}]) \approx \mathbb{E}_{\boldsymbol{Y}|\boldsymbol{X}}[\Psi(\boldsymbol{t}, \boldsymbol{q}, \boldsymbol{p})] = \Psi_{\text{ETU}} \,.$$

- If Ψ is linear in all arguments depending on $\boldsymbol{Y}$ $\implies$ approximation is **exact** (e.g., instance-wise weighted metrics and macro-precision).
- Generally, $\tilde{\Psi}_{\text{ETU}}$ as a surrogate leads only to $\mathcal{O}(1/\sqrt{n})$ error.

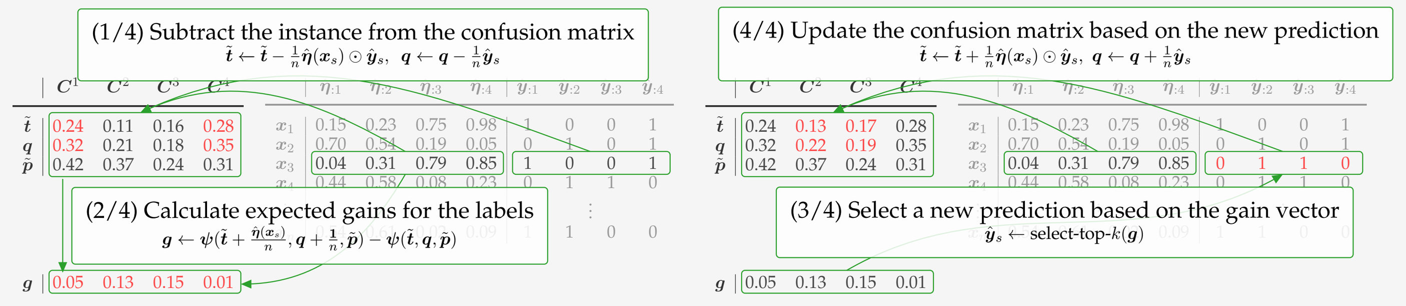
Block Coordinate Ascent (BCA) algorithm

- Optimizing $\tilde{\Psi}_{\text{ETU}}$ can be a very hard discrete optimization problem.
- **Block-coordinate ascent** constructs a sequence of predictions with non-decreasing utility to find local optima.
- Allows for using **sparse** $\boldsymbol{\eta}$ and $\hat{\boldsymbol{Y}}$ to **scale** to XMLC problems.
- We provide **regret bounds** quantifying influence of semi-empirical approximation and label probability estimation error.

Initialization of BCA algorithm:



Repeat for each instance in the random sequence until convergence:



Experimental results

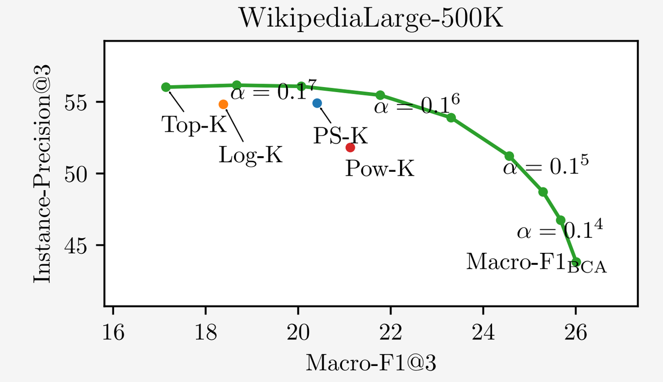
- We compare BCA with popular re-weighting strategies on XMLC benchmarks.
- Results obtained using LIGHTXML as a label-probability estimator:

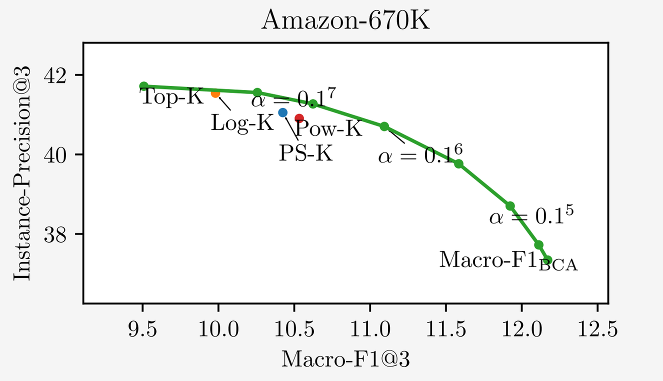
Inference strategy	WikipediaLarge-500K Instance @3 Prec	Instance @3 Rec	Macro @3 Prec	Macro @3 Rec	Macro @3 F1	Macro @3 Cov	Amazon-670K Instance @3 Prec	Instance @3 Rec	Macro @3 Prec	Macro @3 Rec	Macro @3 F1	Macro @3 Cov
TOP-K	**56.0**	**45.7**	20.2	18.9	17.1	32.2	**41.7**	**24.1**	10.8	9.7	9.5	14.4
PS-K	*54.9*	*45.6*	23.3	22.7	20.4	37.8	41.1	23.8	11.9	10.5	10.4	15.5
POW-K	51.8	43.9	23.7	23.7	21.1	39.4	40.9	23.7	12.0	10.7	10.5	15.6
LOG-K	54.8	45.2	21.4	20.2	18.4	34.3	*41.5*	*24.0*	11.3	10.1	10.0	15.0
MACRO-P$_{\text{BCA}}$	25.2	21.8	**37.7**	20.2	23.4	45.1	33.8	19.8	**17.3**	10.5	*12.1*	**17.8**
MACRO-R$_{\text{BCA}}$	43.4	39.6	25.4	**27.6**	*23.7*	46.3	39.4	22.9	13.7	**11.2**	11.5	17.1
MACRO-F1$_{\text{BCA}}$	43.8	36.4	*35.4*	23.7	**26.0**	*46.4*	37.3	21.7	*16.5*	10.8	**12.2**	17.6
COV$_{\text{BCA}}$	27.3	24.6	25.9	*26.8*	21.6	**50.2**	35.4	20.3	14.0	*10.9*	11.2	*17.7*

Interpolated metrics

- Optimizing macro-measures incurs a **significant drop** in instance-wise measures.
- To achieve the desired trade-off between tail and head label performance a straight-forward **combination** between standard instance-wise precision@k, and a marco-average @k metric can be used, e.g.:

$$\Psi(\mathbf{C}(\boldsymbol{Y}, \hat{\boldsymbol{Y}})) = (1-\alpha)\Psi_{\text{Instance-Precision@}k}(\mathbf{C}(\boldsymbol{Y}, \hat{\boldsymbol{Y}})) + \alpha\Psi_{\text{Macro-F1@}k}(\mathbf{C}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}))$$





A combination of a macro-metric and instance precision-at-k can achieve good results on both metrics simultaneously!

NeurIPS '23, December 10–16, 2023, New Orleans, USA