PROPENSITY-SCORED PROBABILISTIC LABEL TREES

Marek Wydmuch[1]   Kalina Jasinska-Kobus[1,2]   Rohit Babbar[3]   Krzysztof Dembczyński[1,4]

[1]Poznan University of Technology, Poland   [2]ML Research at Allegro.pl, Poland   [3]Aalto University, Helsinki, Finland   [4]Yahoo! Research, New York, USA

Motivation

- In modern machine learning applications, the label space can be enormous, containing even millions of different labels (**eXtreme Classification (XC)**):
  - content annotation for multimedia search,
  - different types of recommendation: webpages-to-ads, ads-to-bid-words, users-to-items, queries-to-items, or items-to-queries.
- In these practical applications, **label distribution is often highly imbalanced**, and **relevant labels can be missing**.
- To address this issue, Jain et al. [1] proposed to evaluate XC models in terms of propensity-scored versions of popular measures.

Extreme multi-label classification (XMLC)

- **Multi-label classification**:

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d \xrightarrow{\boldsymbol{h}(\boldsymbol{x})} \boldsymbol{y} = (y_1, y_2, \ldots, y_m) \in \{0,1\}^m$$

- **Positive labels**: $\boldsymbol{x}$ is associated with a subset of labels $\mathcal{L}_{\boldsymbol{x}} \subseteq \mathcal{L}$ (positive labels). Set $\mathcal{L}_{\boldsymbol{x}}$ is identified with the vector $\boldsymbol{y}$, in which $y_j = 1 \Leftrightarrow j \in \mathcal{L}_{\boldsymbol{x}}$.
- **Conditional probability** of label j: $\eta_j(\boldsymbol{x}) = \mathbf{P}(y_j = 1 | \boldsymbol{x}) = \sum_{\boldsymbol{y}: y_j = 1} \mathbf{P}(\boldsymbol{y}|\boldsymbol{x})$
- **Goal**: find a classifier $\boldsymbol{h}(\boldsymbol{x}) : \mathcal{X} \to \mathcal{R}^m$ **minimizing the expected loss**:

$$R_\ell(\boldsymbol{h}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}) \sim \mathbf{P}(\boldsymbol{x},\boldsymbol{y})}(\ell(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})))$$

- **The optimal classifier**: the **Bayes classifier** for a given loss function ℓ is:

$$\boldsymbol{h}^*_\ell = \arg\min_{\boldsymbol{h}} R_\ell(\boldsymbol{h}) \,.$$

Propensity model

- **Correct labeling in case of an extremely large label set is difficult**
  $\Rightarrow$ Common assumption is that positive labels can be missing.
- **Let $\boldsymbol{y}$ be the true** and **$\tilde{\boldsymbol{y}}$ be the observed** label vector such that:

$$\mathbf{P}(\tilde{y}_j = 1 \,|\, y_j = 1) = p_j \,,\ \mathbf{P}(\tilde{y}_j = 0 \,|\, y_j = 1) = 1 - p_j \,,$$
$$\mathbf{P}(\tilde{y}_j = 1 \,|\, y_j = 0) = 0 \,,\ \ \mathbf{P}(\tilde{y}_j = 0 \,|\, y_j = 0) = 1 \,,$$

  where $p_j \in [0,1]$ **is the propensity** of observing a positive label when it is indeed positive (the propensity does not depend on $\boldsymbol{x}$).
- **Both training and test sets do follow the propensity model**.
- The observed conditional probability of label j:

$$\tilde{\eta}_j(\boldsymbol{x}) = \mathbf{P}(\tilde{y}_j = 1 \,|\, \boldsymbol{x}) = p_j \mathbf{P}(y_j = 1 \,|\, \boldsymbol{x}) = p_j \eta_j(\boldsymbol{x}) \,.$$

- The original conditional probability of label j (with **inverse propensity** $q_j = \frac{1}{p_j}$):

$$\eta_j(\boldsymbol{x}) = \mathbf{P}(y_j = 1 \,|\, \boldsymbol{x}) = q_j \mathbf{P}(\tilde{y}_j = 1 \,|\, \boldsymbol{x}) = q_j \tilde{\eta}_j(\boldsymbol{x}) \,.$$

Bayes optimal decisions for psp@k

- **Propensity-scored precision@k** ($psp@k$) [1]:

$$psp@k(\tilde{\boldsymbol{y}}, \boldsymbol{h}_{@k}(\boldsymbol{x})) = \frac{1}{k} \sum_{j \in \hat{\mathcal{L}}_{\boldsymbol{x}}} q_j [\![\tilde{y}_j = 1]\!] \,,$$

  where $\hat{\mathcal{L}}_{\boldsymbol{x}}$ is a set of k labels predicted by $\boldsymbol{h}_{@k}$ for $\boldsymbol{x}$.
- Standard precision@k ($p@k$) is a special case of $psp@k$ if $q_j = 1$ for all j.
- The conditional risk for $\ell_{psp@k} = -psp@k$:

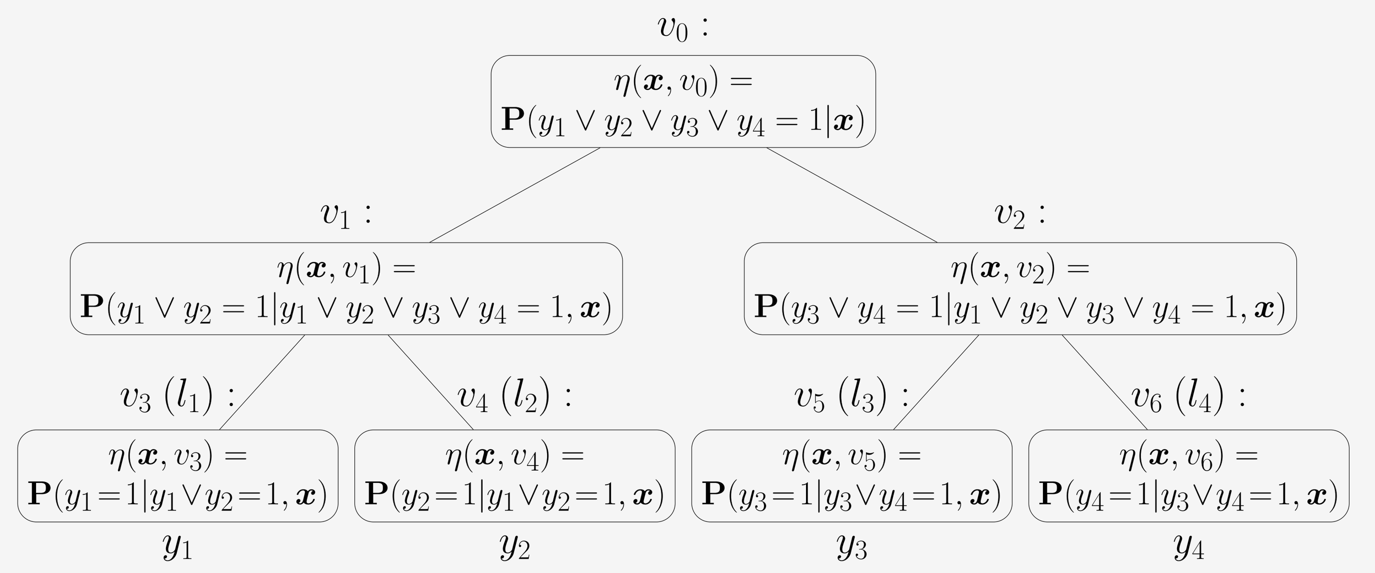
$$R_{psp@k}(\boldsymbol{h}_{@k} \,|\, \boldsymbol{x}) = \mathbb{E}_{\tilde{\boldsymbol{y}}} \ell_{psp@k}(\tilde{\boldsymbol{y}}, \boldsymbol{h}_{@k}(\boldsymbol{x})) = -\frac{1}{k} \sum_{j \in \hat{\mathcal{L}}_{\boldsymbol{x}}} q_j \tilde{\eta}_j(\boldsymbol{x}) \,.$$

- Given propensities or their estimates in the time of prediction, **optimal strategy for $psp@k$: select k labels with the highest values of $q_j \tilde{\eta}_j(\boldsymbol{x})$**.
- **Applying this strategy is not straightforward in case of XMLC**, calculating probability estimates for the full set of labels is not feasible.

Probabilistic Label Trees (PLTs)

- **Probabilistic Label Tree (PLT)** [2] uses a tree, with set of nodes V, in which each leaf $l_j \in L$ corresponds to one label $j \in \mathcal{L}$, to factorize conditional probabilities of labels:

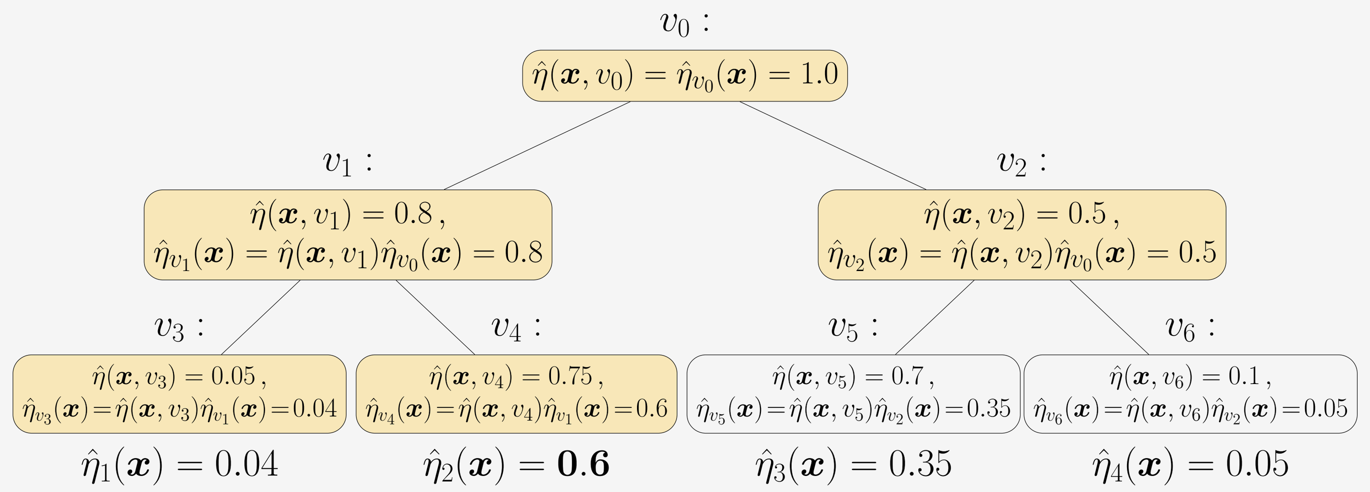
$$\eta_{l_j}(\boldsymbol{x}) = \eta_j(\boldsymbol{x}) = \mathbf{P}(y_j = 1 | \boldsymbol{x}) = \prod_{v \in \mathrm{Path}(y_j)} \eta(\boldsymbol{x}, v)$$



- **PLTs uses binary classifiers in the tree nodes to obtain $\hat{\eta}$ – estimates of η.**
- PLTs has been recently implemented in several state-of-the-art algorithms: PARABEL [3], EXTREMETEXT [4], BONSAI [5], ATTENTIONXML [6], NAPKINXC [7].

Prediction in PLTs

- UNIFORM-COST-SEARCH- or BEAM-SEARCH-based inference can be used to efficiently find k labels with highest estimates of $\eta_j(\boldsymbol{x})$.
- Example of PLT's top-1 inference:



Propensity-scored PLTs (PS-PLTs)

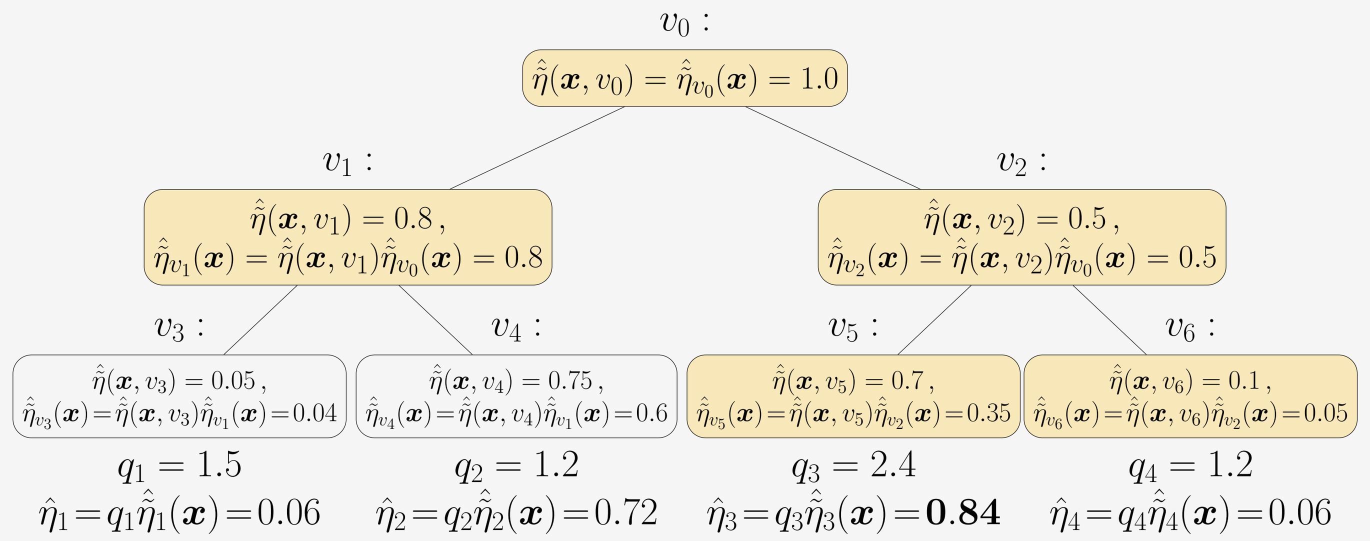
- Since inverse propensities $q_j \geq 1$, we need to introduce a **new A^*-SEARCH-based inference** to find labels with highest values of $q_j \hat{\tilde{\eta}}_j(\boldsymbol{x})$.
- Notice that:

$$q_j \hat{\tilde{\eta}}_j(\boldsymbol{x}) = \exp\Big(-\big(-\log q_j - \sum_{v \in \mathrm{Path}(l_j)} \log \hat{\tilde{\eta}}(\boldsymbol{x}, v) \big) \Big) = \exp\big(-f(l_j, \boldsymbol{x}) \big) \,,$$

  where $f(l_j, \boldsymbol{x})$ is a cost function for label j.
- A^*-SEARCH inference is guided by:

$$\hat{f}(v, \boldsymbol{x}) = g(v, \boldsymbol{x}) + h(v, \boldsymbol{x}) = \overbrace{-\log \max_{j \in \mathcal{L}_v} q_j}^{h(v,\boldsymbol{x})} \overbrace{- \sum_{v' \in \mathrm{Path}(v)} \log \hat{\tilde{\eta}}(\boldsymbol{x}, v')}^{g(v,\boldsymbol{x})} \,,$$

  where $g(v, \boldsymbol{x})$ is a cost of reaching tree node v from the root and $h(v, \boldsymbol{x})$ is a heuristic estimating the cost of reaching the best leaf node from node v.
- PS-PLT inference algorithm is **admissible** and **optimally efficient**.
- Example of PS-PLT's top-1 inference:



Experimental results

- Comparison on benchmark datasets from the XMLC repository [8].
- **True propensities are unknown for the benchmark datasets.**
- **Propensities modeled as proposed by Jain et al. [1]**:

$$p_j = \mathbf{P}(\tilde{y}_j = 1 \,|\, y_j = 1) = \frac{1}{1 + C e^{-A \log(N_j + B)}} \,,$$

  where N_j is the number of data points annotated with label j in the observed ground truth dataset of size N, parameters A and B are specific for each dataset, and $C = (\log N - 1)(B + 1)^A$.
- PS-PLTs compared to SOTA on propensity-scored and standard precision@$\{1,3,5\}$ [%], and on CPU train [h] and prediction times [ms]:

Algorithm	$psp@1$	$psp@3$	$psp@5$	$p@1$	$p@3$	$p@5$	t_{train}	t/N_{test}
	WikipediaLarge-500K, $A = 0.5$, $B = 0.4$							
PROXML [9]	33.10	35.00	**39.40**	**68.80**	**48.90**	37.90	≈1595920	≈496
PW-DISMEC [10]	30.31	31.56	33.52	66.38	45.69	35.85	≈16272	≈457
PFASTREXML [1]	29.20	27.60	27.70	59.50	40.20	30.70	51.07	15.24
PARABEL [3]	28.80	31.90	34.60	67.50	48.70	37.70	7.83	3.84
PLT [7]	26.11	30.76	33.98	67.48	48.19	37.65	62.39	14.58
PS-PLT (ours)	**33.69**	**35.34**	37.63	67.52	48.71	**38.09**		30.02
	Amazon-670K, $A = 0.6$, $B = 2.6$							
PROXML	**30.80**	32.80	35.10	43.50	38.70	35.30	≈75160	≈111
PW-DISMEC	30.60	**33.27**	**35.51**	41.70	37.81	34.92	≈810	≈103
PFASTREXML	29.30	30.80	32.43	39.46	35.81	33.05	3.01	9.96
PARABEL	25.43	29.43	32.85	**44.89**	**39.80**	36.00	0.46	1.73
PLT	26.01	29.80	33.31	44.47	39.73	**36.25**	1.92	5.25
PS-PLT	30.67	32.94	34.96	43.25	39.28	36.06		9.56

Source code: `https://github.com/mwydmuch/napkinXC`

References

[1] Himanshu Jain, Yashoteja Prabhu, and Manik Varma. Extreme multi-label loss functions for recommendation, tagging, ranking and other missing label applications. KDD, 2016

[2] Kalina Jasinska, Krzysztof Dembczynski, Róbert Busa-Fekete, Karlson Pfannschmidt, Timo Klerx, and Eyke Hüllermeier. Extreme F-measure maximization using sparse probability estimates. ICML, 2016

[3] Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *WWW*, 2018

[4] Marek Wydmuch, Kalina Jasinska, Mikhail Kuznetsov, Róbert Busa-Fekete, and Krzysztof Dembczynski. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In *NeurIPS*, 2018

[5] Sujay Khandagale, Han Xiao, and Rohit Babbar. Bonsai - diverse and shallow trees for extreme multi-label classification. 2019

[6] Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. AttentionXML: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In *NeurIPS*. 2019

[7] Kalina Jasinska-Kobus, Marek Wydmuch, Krzysztof Dembczyński, Mikhail Kuznetsov, and Róbert Busa-Fekete. Probabilistic label trees for extreme multi-label classification. 2020

[8] Kush Bhatia, Kunal Dahiya, Himanshu Jain, Anshul Mittal, Yashoteja Prabhu, and Manik Varma. The extreme classification repository: Multi-label datasets and code, 2016

[9] Rohit Babbar and Bernhard Schölkopf. Data scarcity, robustness and extreme multi-label classification. *Machine Learning*, 108, 2019

[10] Mohammadreza Qaraei, Erik Schultheis, Priyanshu Gupta, and Rohit Babbar. Convex surrogates for unbiased loss functions in extreme classification with missing labels. In *WWW*, 2021