Extreme Multi-label Loss Functions for Recommendation, Tagging, Ranking & Other Missing Label Applications

Himanshu Jain HIMANSHU.J689@GMAIL.COM
Yashoteja Prabhu YASHOTEJA.PRABHU@GMAIL.COM
IIT Delhi

Manik Varma MANIK@MICROSOFT.COM
Microsoft Research

Abstract

The choice of the loss function is critical in extreme multi-label learning where the objective is to annotate each data point with the most relevant *subset* of labels from an extremely large label set. Unfortunately, existing loss functions, such as the Hamming loss, are unsuitable for learning, model selection, hyperparameter tuning and performance evaluation. This paper addresses the issue by developing propensity scored losses which: (a) prioritize predicting the few relevant labels over the millions of irrelevant ones; (b) do not erroneously treat missing labels as irrelevant; and (c) promote the accurate prediction of hard to predict, but rewarding tail labels. Another contribution is the development of algorithms which efficiently scale to extremely large datasets with up to 9 million labels, 70 million points and 2 million dimensions and which give significant improvements over the state-of-the-art. We also demonstrate that the proposed contributions achieve superior clickthrough rates on sponsored search ranking problems in Bing.

1. Introduction

Extreme multi-label learning (XML) addresses the problem of learning a classifier that can annotate a data point with the most relevant *subset* of labels from an extremely large label set. XML is an important research problem because many applications like tagging, recommendation and ranking can naturally be reformulated as XML tasks. For example in e-retailing, each product can be treated as a separate label, followed by learning an XML classifier that maps a user's feature vector to a set of relevant labels (products), and then using the classifier to predict the subset of products that a new user might like to purchase.

The choice of an appropriate loss function is critical to ensure successful training, hyper-parameter tuning, model selection and performance evaluation in an XML task. Traditional multi-label loss functions like Hamming loss are rendered unsuitable when applied to XML tasks due to following reasons: First, data points in an XML task naturally have many missing labels in their ground truth since it is impossible to accurately judge each instance against millions of labels. Traditional loss functions erroneously treat the missing labels as irrelevant. Second, they do not prioritize predicting the few relevant labels over the millions of irrelevant ones. Furthermore, the infrequently occurring tail labels are often more informative and rewarding while being harder to predict than frequently occurring ones. Traditional loss functions treats all relevant labels as being equally important.

The primary contribution of this paper is to develop loss functions suitable for extreme multi-label learning. Propensity scored variants of ranking losses like precision@k and nDCG@k, are developed and proved to give unbiased estimates of the true loss function even when ground truth labels go missing under arbitrary probabilistic label noise models. This is shown to naturally promote more rewarding tail labels. Another contribution is the development of an efficient extreme multi-label algorithm that can scale to extremely large datasets with up to 9 million labels, 70 million training points and 2 million dimensional features and achieves significant improvements over the state-of-the-art.

Preliminary work. Under review by the Indian Workshop on Machine Learning (iWML). Do not distribute.

2. Related Work

Although some algorithms have been proposed for training with missing labels (Yu et al., 2014; Kong et al., 2014) under restrictive settings, aspects such as hyper-parameter tuning, model selection and performance evaluation have not been addressed before. As such, the Hamming loss (Karampatziakis & Mineiro, 2015) continues to be one of the most popular losses for extreme multi-label learning along with precision (Prabhu & Varma, 2014; Karampatziakis & Mineiro, 2015) and the F-measure (Karampatziakis & Mineiro, 2015). On the other hand, unbiased estimators for recall (Steck, 2010) and average discounted gain have been developed under the restrictive assumption that labels go missing uniformly at random from the ground truth. By contrast, this paper develops propensity scored variants of precision, nDCG and other loss functions and proves that they are unbiased even under general probabilistic label noise models.

Propensity scoring has been used to develop unbiased estimators for observational data (Rosenbaum & Rubin, 1983).

Heuristical label (item) weighting loss functions have been proposed to promote the accurate prediction of infrequently occurring labels (rare items) which might delight and surprise the user (Vargas & Castells, 2011). This paper provides theoretical justification for such heuristics by showing how similar weights can arise from the proposed propensity models.

3. Propensity Scored Losses

This Section develops propensity scored variants of precision@k, nDCG@k and other popular loss functions, which are computed on the observed labels and provide unbiased estimates of the true loss function computed on the complete (but unobtainable) ground truth without any missing labels.

Let $\boldsymbol{y^*}, \boldsymbol{y} \in \{0,1\}^L$ denote the complete (but unobtainable) and observed (but with missing labels) ground truth label vectors for a given data point such that $y^*_l = y_l = 1$ for observed relevant labels, $y^*_l = 1, y_l = 0$ for unobserved relevant labels and $y^*_l = y_l = 0$ for irrelevant labels.

Let $p_{il} \equiv P(y_{il} = 1 | y^*_{il} = 1)$ denote the propensity, that is the marginal probability of a relevant label l being observed for a data point i.

Let $\mathcal{L}^*(\boldsymbol{y^*}, \hat{\boldsymbol{y}}) = \sum_{l=1}^{L} \mathcal{L}^*_l(y^*_l, \hat{y}_l) = \sum_{l:y^*_l=1}^{L} \mathcal{L}^*_l(1, \hat{y}_l)$ denote the family of loss functions which decompose over individual labels l and are computed over the relevant labels alone ($\{l | y^*_l = 1\}$). $\mathcal{L}^*$ represents the true loss function measuring the loss incurred for predicting $\hat{\boldsymbol{y}}$ when the complete ground truth vector was $\boldsymbol{y^*}$. The propensity scored variant of $\mathcal{L}^*$ computed on the observed ground truth $\boldsymbol{y}$ is defined to be $\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{l:y_l=1}^{L} \mathcal{L}_l(1, \hat{y}_l) = \sum_{l:y_l=1}^{L} \mathcal{L}^*_l(1, \hat{y}_l)/p_l$.

Then the following theorem implies that $\mathcal{L}$ can be a viable proxy for $\mathcal{L}^*$ for training, model selection, hyperparameter tuning and performance evaluation.

Theorem 3.1. *The loss function $\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ evaluated on the observed ground truth $\boldsymbol{y}$ is an unbiased estimator of the true loss function $\mathcal{L}^*(\boldsymbol{y^*}, \hat{\boldsymbol{y}})$ evaluated on complete ground truth $\boldsymbol{y^*}$. Thus, $\mathbb{E}_{\boldsymbol{y}}[\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}})] = \mathbb{E}_{\boldsymbol{y^*}}[\mathcal{L}^*(\boldsymbol{y^*}, \hat{\boldsymbol{y}})]$, for any $P(\boldsymbol{y^*})$ and $P(\boldsymbol{y})$ related through propensities p_l and any fixed $\hat{\boldsymbol{y}}$.*

Proof. For proof, please refer to full version. □

4. Propensity Model

Section 3 requires that the marginal propensities of labels being retained is known. Unfortunately, propensities are generally unknown as $\boldsymbol{y^*}$ is unavailable due to the large label space. Based on empirical observation, this Section proposes that the propensities can be modelled as a sigmoidal function of $\log N_l$

$$p_l \equiv P(y_l = 1 | y^*_l = 1) = \frac{1}{1 + Ce^{-A \log(N_l + B)}} \qquad (1)$$

where N_l is the number of data points annotated with label l in the observed ground truth dataset of size N and A, B, C are model parameters. In particular, propensities are estimated on Wikipedia and Amazon where meta-data is available for the task and shown to give a close fit to (1) (see Figure 1).

For the empirical estimation of propensities, we make use of the category hierarchy and ”items viewed together” information, respectively, in Wikipedia and Amazon datasets. For more details, please refer to full version.

5. Algorithms

This Section develops the PfastreXML algorithm for extreme multi-label learning. PfastreXML optimizes propensity scored nDCG by leveraging FastXML (Prabhu & Varma, 2014) for nDCG optimization. PfastreXML then further extends FastXML to improve the tail label prediction which is the most challenging aspect of extreme multi-label learning. PfastreXML makes key approximations which increase FastXML's training time by just seconds while retaining the prediction accuracy gains of the extension.

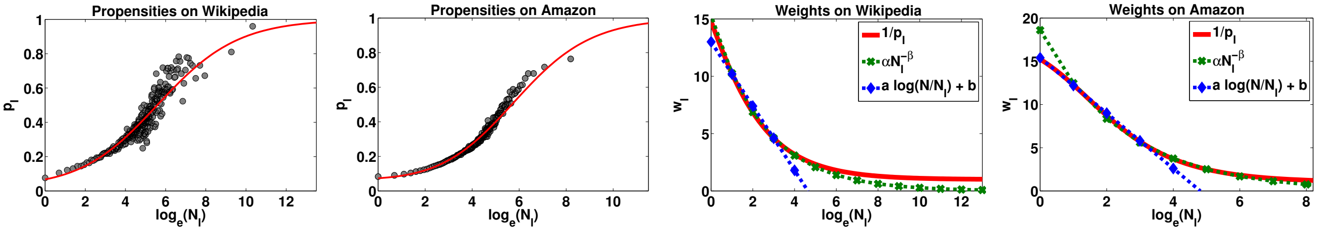

Figure 1. Propensities p_l and their corresponding weights $w_l = 1/p_l$ on Wikipedia and Amazon. The estimated propensities follow a sigmoidal curve on the semi-log plot and provide a principled setting of the weights for recommending rare items as compared to popular heuristics such as $N_l^{-\beta}$ and $\log(N/N_l)$.

5.1. Propensity scored FastXML

Propensity scored FastXML (PfastXML) shares the same architecture as FastXML (Prabhu & Varma, 2014) which learns an ensemble of trees during training. PfastXML improves upon FastXML by replacing the nDCG loss with its propensity scored variant which is unbiased and assigns higher rewards for accurate tail label predictions. In particular, PfastreXML replaces $\mathcal{L}_{\text{nDCG}@L}$ in (Prabhu & Varma, 2014) by $\mathcal{L}_{\text{PSnDCG}@L}(\boldsymbol{r},\boldsymbol{y}) = -\frac{\sum_l \frac{y_l}{p_l \log(r_l+1)}}{\sum_{l=1}^{L} \frac{1}{\log(1+l)}}$

FastXML's objective function can be recovered from PfastXML's by substituting $y_{il}^p = y_{il}/p_{il}$, and can therefore be optimized FastXML's iterative alternating optimization applied to y_{il}^p.

PfastXML enjoys all the scaling properties of FastXML, while improving prediction accuracy.

5.2. PfastreXML

Propensity scoring improves FastXML but tree classifiers are still prone to predicting tail labels with low probabilities. PfastreXML addresses this limitation by re-ranking PfastXML's predictions using classifiers designed specifically for tail labels.

Each tail label has little training data which is limited to certain regions of feature space. Compact hyperspherical decision boundaries are therefore learnt for each tail label independently according to:

$$P(y_{il}^*|\boldsymbol{x_i}) = 1/(1 + v_{il}^{2y_{il}^*-1}) \tag{2}$$

$$\text{where} \quad v_{il} = e^{\frac{\gamma}{2}\|\boldsymbol{x_i}-\boldsymbol{\mu_l}\|_2^2}$$

For optimizing the above objective, instead of a slower gradient descent based approach, we propose a very fast and approximate solution. Assuming that the relevant labels lie within a very tight cluster leads to $u_{il} \approx y_{il}$ yielding

$$\boldsymbol{\mu_l^*} = \frac{\sum_{i=1}^{N} y_{il}\boldsymbol{x_i}}{\sum_{i=1}^{N} y_{il}} \tag{3}$$

Thus, each $\boldsymbol{\mu_l^*}$ turns out to be the sparse mean of the training points for which the label was observed to be relevant.

Re-ranking: The final ranked list of labels, restricted to the label set predicted by PfastXML, is obtained by sorting the weighted average of probability scores from PfastXML and 2 obtained as follows:

$$s_l = \alpha \log P_{\text{pf}}(y_l^* = 1|\boldsymbol{x}) + (1-\alpha)\log P(y_l^* = 1|\boldsymbol{x}) \tag{4}$$

6. Experiments

Experiments were carried out on the largest benchmark datasets demonstrating that PfastreXML could achieve significantly higher prediction accuracies according to the unbiased propensity scored loss functions as compared to the state-of-the-art.

Datasets: We used several extreme multi-label datasets including Ads-9M, WikiLSHTC-325K, Amazon-670K, EUR-Lex and a few others.

Baseline algorithms: Our baselines include FastXML (Prabhu & Varma, 2014) and SLEEC (Bhatia et al., 2015) which are the leading tree and embedding based approaches respectively. Other baseline algorithms include 1-vs-All, LEML, WSABIE, CPLST, CS, ML-CSSP, LPSR, and a popularity baseline. For references to these datasets and baselines, click here.

Evaluation metrics: Performance is evaluated using the unbiased propensity scored Precision@k and nDCG@k, normalized to lie between $\{0, 1\}$.

Results: Table 5 compares PfastreXML's performance to that of state-of-the-art SLEEC, FastXML and other baseline algorithms using unbiased precision. Most of the baseline algorithms do not scale to large datasets and hence their corresponding results are missing. As can be seen, the proposed PfastXML and PfastreXML lead to significantly better prediction accuracies as compared to the state-of-the-art.

Table 1. The proposed PfastreXML and PfastXML algorithms make significantly more accurate predictions as compared to state-of-the-art SLEEC, FastXML and other baseline algorithms. Performance is evaluated according to the unbiased propensity scored Precision@k (PK) and nDCG@k (Nk) for $k = 1, 3$ and 5.

(a) EUR-Lex $N = 15K, D = 5K, L = 4K$

Algorithm	N1(%)	N3(%)	N5(%)	P1(%)	P3(%)	P5(%)
Popularity	1.80	2.10	2.36	1.80	2.20	2.62
1-vs-All	37.97	42.44	43.97	37.97	44.01	46.17
SLEEC	35.45	39.79	41.97	35.45	41.35	44.62
LEML	24.33	26.45	27.70	24.33	27.22	29.13
WSABIE	31.65	34.12	35.43	31.65	35.04	36.99
CPLST	28.93	31.60	32.92	28.93	32.57	34.55
CS	25.31	26.98	25.71	25.31	27.57	25.13
ML-CSSP	25.25	26.70	27.79	25.25	27.27	28.97
FastXML	27.61	33.22	36.28	27.61	35.35	39.95
LPSR	33.65	38.20	39.82	33.65	39.88	42.17
PfastXML	41.31	44.01	45.13	41.31	45.02	46.67
PfastreXML	**45.38**	**46.42**	**47.25**	**45.38**	**46.79**	**48.08**

(d) WikiLSHTC-325K $N = 1.78M, D = 1.62M, L = 325K$

Algorithm	N1(%)	N3(%)	N5(%)	P1(%)	P3(%)	P5(%)
Popularity	2.56	1.91	1.83	2.56	1.65	1.53
SLEEC	20.51	22.45	23.52	20.51	23.32	25.23
FastXML	16.52	19.70	21.17	16.52	21.12	23.69
PfastXML	25.58	26.55	27.42	25.58	27.01	28.59
PfastreXML	**31.16**	**31.56**	**32.40**	**31.16**	**31.80**	**33.35**

(e) Amazon-670K $N = 490K, D = 136K, L = 670K$

Algorithm	N1(%)	N3(%)	N5(%)	P1(%)	P3(%)	P5(%)
Popularity	0.03	0.04	0.04	0.03	0.04	0.04
SLEEC	20.62	22.63	24.43	20.62	23.32	25.98
FastXML	20.20	22.94	25.26	20.20	23.88	27.28
PfastXML	25.61	26.95	28.09	25.61	27.42	29.09
PfastreXML	**29.93**	**30.91**	**31.94**	**29.93**	**31.26**	**32.80**

(f) Ads-9M $N = 70.45M, D = 2.08M, L = 8.84M$

Algorithm	N1(%)	N3(%)	N5(%)	P1(%)	P3(%)	P5(%)
Popularity	0.05	0.08	0.09	0.05	0.09	0.12
FastXML	6.18	6.72	6.94	6.18	6.99	7.42
PfastXML	6.60	7.10	7.32	6.60	7.37	7.76
PfastreXML	**8.75**	**9.87**	**10.28**	**8.75**	**10.45**	**11.20**

PfastreXML also improves upon PfastXML's prediction accuracy with negligible training and prediction overhead ($\sim$ 18 min, and 0.13 ms, respectively, on Ads-9M).

PfastreXML's query rankings were also used to serve ads on the Bing search engine, and was observed to give an improvement of significantly more than 5% in the clickthrough rate over the existing system. This helps verify that the propensity scored loss functions and proposed algorithm align with the requirements of real world applications.

7. Conclusions

This paper developed loss functions suitable for extreme multi-label learning and long tail, missing label applications such as ranking, recommendation and tagging. Propensity scored variants of precision and nDCG were developed and proved to give unbiased estimates of the true loss function evaluated on the complete ground truth, as well as promote the accurate prediction of tail labels.

We also developed the PfastreXML algorithm for optimizing propensity scored nDCG. PfastreXML was shown to make significantly more accurate predictions on all datasets as compared to state-of-the-art XML classifiers, and achieve significantly higher click-through rates for sponsored search advertising on Bing as compared to the existing system.

Dual Submissions

This work is currently under submission to 22nd ACM SIGKDD Conference, to be held in San Francisco, California in Aug-2016. To read the full version submitted to KDD, click here.To visit the conference website, click here.

References

Bhatia, K., Jain, H., Kar, P., Varma, M., and Jain, P. Sparse local embeddings for extreme multi-label classification. In *NIPS*, 2015.

Karampatziakis, N. and Mineiro, P. Scalable multi-label prediction via randomized methods. *CoRR*, 2015. URL `http://arxiv.org/abs/1502.02710`.

Kong, X., Wu, Z., Li, L. J., Zhang, R., Yu, P. S., Wu, H., and Fan, W. Large-scale multi-label learning with incomplete label assignments. In *SDM*, 2014.

Prabhu, Y. and Varma, M. FastXML: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *KDD*, 2014.

Rosenbaum, P. R. and Rubin, D. B. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 1983.

Steck, H. Training and testing of recommender systems on data missing not at random. In *KDD*, 2010.

Vargas, S. and Castells, P. Rank and relevance in novelty and diversity metrics for recommender systems. In *RecSys*, 2011.

Yu, H. F., Jain, P., Kar, P., and Dhillon, I. S. Large-scale multi-label learning with missing labels. In *ICML*, 2014.