A Revisit of eXtreme Multi-label Learning

Tong Wei

December 12, 2018

Contents

1 Testing Time Speedup

- **[column subset selection]** Boutsidis *et al.* [2009]
  They propose to find approximate solutions of the Column Subset Selection Problem (CSSP) more efficiently.

- **[label selection]** Bi and Kwok [2013]
  They address this problem by selecting a small subset of class labels that can approximately span the original label space. This is performed by an efficient randomized sampling procedure where the sampling probability of each class label reflects its importance among all the labels.

- **[label partition]** Weston *et al.* [2013]
  It works by first partitioning the input space, so any given example can be mapped to a partition or set of partitions. In each partition only a subset of labels is considered for scoring by the given label scorer.

- **[label filter]** Niculescu-Mizil and Abbasnejad [2017]
  They propose a two step approach where computationally efficient label filters pre-select a small set of candidate labels before the base multi-class or multi-label classifier is applied.

- **[block-wise partition]** Liang *et al.* [2018]
  They propose a Block-wise Partitioning (BP) pretreatment that divides all instances into disjoint clusters, to each of which the most frequently tagged label subset is attached. One multi-label classifier is trained on one pair of instance and label clusters, and the label set of a test instance is predicted by first delivering it to the most appropriate instance cluster.

- **[structure prediction & ECOC]** Evron *et al.* [2018]
  This work is based on Jasinska and Karampatziakis [2016] which can be seen as a special case of Error-Correcting Output Coding (ECOC). In addition to the logarithmic inference time and model size benefiting from the trellis graph, the authors introduce theoretical bounds for their methods follow previous work on EOOC. Interestingly, both Jasinska and Karampatziakis [2016] and One-Vs-Rest (OVR) can be seen as special cases of the proposed approach.

2 Training Time Speedup

- **[parallelization]** Babbar and Schölkopf [2017]
  They propose a large-scale distributed framework for learning one-vs-rest

linear classifiers coupled with explicit capacity control to control model size. By employing a double layer of parallelization, it can gain significant training speedup over SLEEC and other SOTA.

- **[structure prediction on trellis]** Jasinska and Karampatziakis [2016]
  The authors construct a directed acyclic graph (DAG) G with $O(\log_2 L)$ edges that contains exactly L (number of labels/classes) paths from a source vertex to a sink vertex. Every edge e in the graph is associated with a learnable function. Every class corresponds to a path and the model predicts the class with the highest scoring path.

3 Model Size Reduction

- **[spurious parameters removing]** Babbar and Schölkopf [2017]
  They propose a large-scale distributed framework for learning one-vs-rest linear classifiers coupled with explicit capacity control to control model size. The experiments find that the majority of model parameters are close to 0 and can be filtered out.

- **[regularizor]** Yen *et al.* [2016]
  They show that a margin-maximizing loss with ℓ_1 penalty, in case of Extreme Classification, yields extremely sparse solution both in primal and in dual without sacrificing the expressive power of predictor.

4 Tail Label

- **[low-rank + sparse]** Xu *et al.* [2016]
  They propose to decompose the label matrix into a low-rank matrix and a sparse one. The low-rank matrix is expected to capture the correlation between labels and the sparse one is employed to capture tail labels.

- **[low-rank + sparse]** Li *et al.* [2017]
  They decompose the user-item matrix into low-rank and sparse components.

- **[hamming loss]** Babbar and Schölkopf [2018]
  They detect tail labels by optimizing hamming-loss and designs a robust framework to model data scarcity of tail labels.

- **[propensity score]** Jain *et al.* [2016]
  They propose propensity model that promotes the accurate prediction of infrequent labels with high ranks.

- **[meta learning]** Wang and Hebert [2016]
  They learn a meta-level network that operates on the space of model parameters, which is specifically trained to regress many-shot model parameters (trained on large data sets) from few-shot model parameters (trained on small data sets).

- **[transfer learning]** Wang *et al.* [2017]
  They cast the long tail classification problem as transfer learning, where knowledge from the data-rich classes in the head of the distribution is transferred to the data-poor classes in the tail.

5 Missing Label

- **[ignore]** Yu *et al.* [2014]
  They handle missing labels by training model on observed labels only which means the position of missing entries in label matrix need be known in advance. Such formulation has elegant theoretical analysis, however, can not capture tail label practically.

- **[metric based]** Jain *et al.* [2016]
  They does not erroneously treat missing labels as irrelevant but instead provide unbiased estimates of the true loss function even when ground truth labels go missing under arbitrary probabilistic label noise models. This paper addresses this issue by developing propensity scored variants of precision@k and nDCG@k which provide unbiased estimates of the true loss as if computed on the complete ground truth without any missing labels.

- **[metric based]** Prabhu *et al.* [2018]
  The above two metric based methods, designing unbiased loss functions for XML even when labels are not fully revealed. However, missing labels in training data can not be predicted.

- **[false-negativeness approximation]** Kanehira *et al.* [2016]
  In XML, there are many false-negative examples which may severely degrade the performance when using AUC as the optimization objective. The authors train an uni-class model and approximate false-negativeness of each examples for each label. Then use false-negativeness as another penalty term in the objective. Overall, this work is somewhat incremental, but it brings a possible method to deal with false-negative examples.

- **[propensity score]** Yang *et al.* [2018]
  In recommender systems, since only positive feedback are observed, the authors prove the evaluation metric and the recommendation algorithms are biased toward popular items. The authors then propose an unbiased estimator using inverse propensity score.

6 General XML Methods

6.1 Embedding-based Methods

- **[low-rank]** Yu *et al.* [2014]
  They take a direct approach by formulating the XML problem as that of learning a low-rank linear model. Unlike former embedding based approaches which attempt to make training and prediction tractable by assuming that the training label matrix is low-rank and reducing the effective number of labels by projecting the high dimensional label vectors onto a low dimensional linear subspace.

- **[piecewise low-rank]** Bhatia *et al.* [2015]
  They learn a small ensemble of local distance preserving embeddings which can accurately predict infrequently occurring (tail) labels. This allows SLEEC to break free of the traditional low-rank assumption and boost classification accuracy by learning embeddings which preserve pairwise distances between only the nearest label vectors.

- **[autoencoder]** Yeh *et al.* [2017]
  They perform joint feature and label embedding by deriving a deep latent space, followed by the introduction of label-correlation sensitive loss function for recovering the predicted label outputs.

- **[joint label and feature embedding in two steps]** Zhang *et al.* [2017]
  They explore the label space by building and modeling an explicit label graph and learn non-linear embedding for both feature and label space.

- **[knn embedding]** Tagami [2017]
  They present a novel graph embedding method called "AnnexML". At the training step, AnnexML constructs a knn graph of label vectors and attempts to reproduce the graph structure in the embedding space. The prediction is efficiently performed by using an approximate nearest neighbor search method that efficiently explores the learned k-nearest neighbor graph in the embedding space.

Embedding methods have many advantages including simplicity, ease of implementation, strong theoretical foundations, the ability to handle label correlations, the ability to adapt to online and incremental scenarios, etc. Unfortunately, embedding methods can also pay a heavy price in terms of prediction accuracy due to the loss of information during the compression phase. For instance, none of the embedding methods developed so far have been able to consistently outperform the 1-vs-rest baseline.

6.2 Tree-based Methods

- **[random forest]** Agrawal *et al.* [2013]
  They develop Multi-label Random Forests to tackle problems with millions of labels.

- **[tree optimizing nDCG]** Prabhu and Varma [2014]
  They formulate a novel node partitioning objective which directly optimizes an nDCG based ranking loss and which implicitly learns balanced partitions.

- **[recall tree]** Daume III *et al.* [2016]
  They create a new online reduction of multi-class classification to binary classification for which training and prediction time scale logarithmically with the number of classes. They use an OAA-like structure to make a final prediction, but instead of scoring every class, we only score a small subset of $O(\log K)$ classes by dynamically builting tree to efficiently whittle down the set of candidate classes. The goal of the tree is to maximize the recall of the candidate set.

- **[gbdt]** Si *et al.* [2017]
  They show that vanilla GBDT can easily run out of memory or encounter near-forever running time in the XML setting, and propose a new GBDT variant, GBDT-SPARSE, to resolve this problem by employing L_0 regularization. They make the crucial observation that each data point has very few labels; based on that we solve a L_0 regularized optimization problem to enforce the prediction of each leaf node in each tree to have only a small number (k) of nonzero elements or labels. Hence, after T trees have been added during GBDT iterations, there will be at most Tk nonzero gradients for any data point.

- **[random forest]** Siblini *et al.* [2018]
  (i) It exploits a random forest strategy which not only randomly reduces both the feature and the label spaces to obtain diversity but also replaces random

selections with random projections to preserve more information; (ii) it uses a novel low-complexity splitting strategy which avoids the resolution of a multi-objective optimization problem at each node.

Table 1: A summary of the advantages and disadvantages of XML methods. ✓ and ✗ indicate a significant superiority and inferiority to other methods respectively. Fields are left blank if the corresponding method could be adapted to deal with that scenario but is not able to achieve outstanding performance.

Method \ Metric	Training time	Testing time	Model size	Predictive Accuracy	Tail label	Missing label
Embedding-based			✓			
Tree-based		✓				
Binary Relevance	✗	✗	✗	✓		

7 Commonly Used XML Performance Measures

7.1 P@k

Top-k precision is a commonly used ranking based performance measure in XML and has been widely adopted for ranking tasks [Prabhu and Varma, 2014; Bhatia *et al.*, 2015]. In Top-k precision, only a few top predictions of an instance will be considered. For each instance $\mathbf{x}_i$, the Top-k precision is defined for a predicted score vector $\hat{\mathbf{y}}_i \in \mathcal{R}^L$ and ground truth label vector $\mathbf{y}_i \in \{-1, 1\}^L$ as

$$\text{P@}k := \frac{1}{k} \sum_{l \in \text{rank}_k(\hat{\mathbf{y}})} \mathbf{y}_l, \tag{1}$$

where $\text{rank}_k(\hat{\mathbf{y}}_i)$ returns the indices of k largest value in $\hat{\mathbf{y}}_i$ ranked in descending order.

7.2 nDCG@k

nDCG@k is another commonly used ranking based performance measure and is defined as

$$\text{nDCG@}k := \frac{\text{DCG@}k}{\sum_{l=1}^{\min(k, \|\mathbf{y}\|_0)} \frac{1}{\log(l+1)}}, \tag{2}$$

where $\text{DCG@}k := \sum_{l \in \text{rank}_k(\hat{\mathbf{y}})} \frac{\mathbf{y}_l}{\log(l+1)}$ and $||\mathbf{y}||_0$ returns the 0-norm of the true-label vector.

7.3 PSP@k

Propensity scored variants of such losses, including precision@k and nDCG@k, are developed and proved to give un- biased estimates of the true loss function even when ground truth labels go missing under arbitrary probabilistic label noise models [Jain *et al.*, 2016].

$$\text{PSP@}k := \frac{1}{k} \sum_{l \in \text{rank}_k(\hat{\mathbf{y}})} \frac{\mathbf{y}_l}{p_l} \tag{3}$$

p_l is the propensity score for label l which helps in making metrics unbiased.

7.4 PSnDCG@k

$$\text{PSDCG@}k := \sum_{l \in \text{rank}_k(\hat{\mathbf{y}})} \frac{\mathbf{y}_l}{p_l \log(l+1)} \tag{4}$$

where $\text{PSnDCG@}k := \frac{\text{PSDCG@}k}{\sum_{l=1}^{k} \frac{1}{\log(l+1)}}$

8 Dig into the Data

8.1 Dataset Statistics

The detail statistics of commonly used XML datasets are listed in Table 2.

Table 2: Data sets statistics

Data set	Train N	Features D	Labels L	Test M	Avg. labels per point	Avg. points per label
Bibtex	4,880	1,836	159	2,515	2.40	111.71
Delicious	12,920	500	983	3,185	19.03	311.61
EUR-Lex	15,539	5,000	3,993	3,809	5.31	25.73
Wiki10	14,146	101,938	30,938	6,616	18.64	8.52
DeliciousLarge	196,606	782,585	205,443	100,095	75.54	72.29
WikiLSHTC-325K	1,778,351	1,617,899	325,056	587084	17.4	3.2
Wiki-500K	1,813,391	2,381,304	501,070	783743	24.7	4.7
Amazon-670K	490,499	135,909	670,091	153025	3.9	5.4

8.2 Raw Feature & Label of Dataset

8.2.1 Amazon Dataset

On Amazon dataset, each instance represents an item (usually a book) identified with an unique item id.

Raw Instance Feature: The raw instance feature are website contents (usually, the instance feature we use in experiments except in deep learning are processed using NLP techniques, such as one-hot encoding), assume the items are books, including book id, title, author, consumer reviews and other informations. The webpage of a book on Amazon is shown in Figure 1.

Raw Label: The meta-label of this web page is the categories that this page belongs to, such as **Politics, Social Sciences, Politics, Government**.

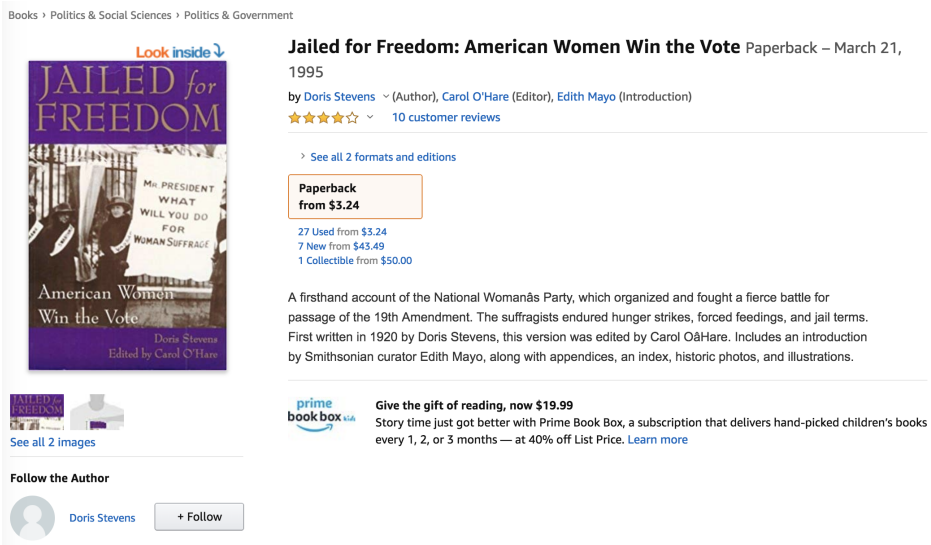

Figure 1: Webpage of an instance on Amazon dataset.

8.2.2 Wikipedia Dataset

On Wikipedia dataset, each instance represents a web page on Wikipedia web site.

Raw Instance Feature: The raw instance feature are website contents. The webpage about "PHP" is shown in Figure 2.

Raw Label: The meta-label of this web page is the categories that this page belongs to, such as **programming language, PHP**.

References

R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings*

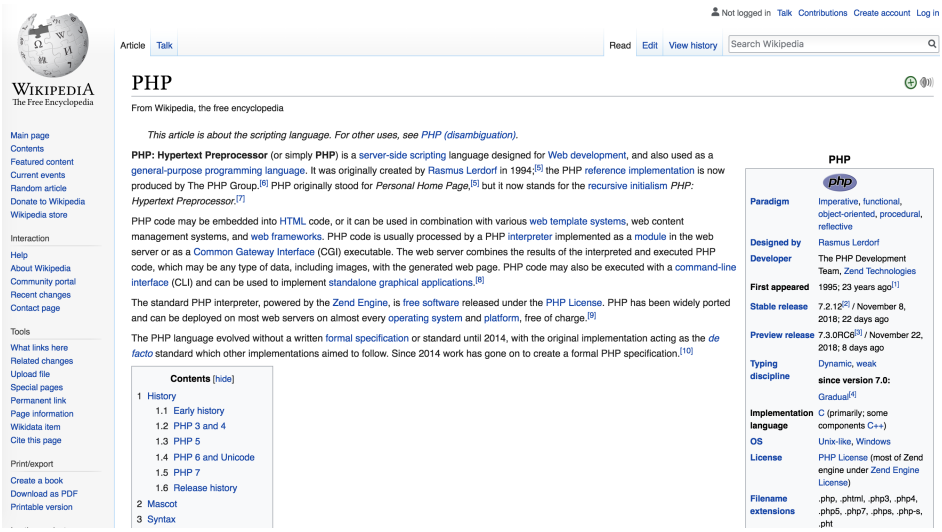

Figure 2: Webpage of an instance on Wikipedia dataset.

of the 22nd international conference on World Wide Web, pages 13–24. ACM, 2013.

R. Babbar and B. Schölkopf. Dismec: Distributed sparse machines for extreme multi-label classification. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, pages 721–729, Cambridge, UK, 2017.

R. Babbar and B. Schölkopf. Adversarial extreme multi-label classification. *arXiv preprint arXiv:1803.01570*, 2018.

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*, pages 730–738, Montreal, Canada, 2015.

W. Bi and J. Kwok. Efficient multi-label classification with many labels. In *Proceedings of International Conference on Machine Learning*, pages 405–413, 2013.

C. Boutsidis, M. W. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 968–977. Society for Industrial and Applied Mathematics, 2009.

H. Daume III, N. Karampatziakis, J. Langford, and P. Mineiro. Logarithmic time one-against-some. *arXiv preprint arXiv:1606.04988*, 2016.

I. Evron, E. Moroshko, and K. Crammer. Efficient Loss-Based Decoding on Graphs For Extreme Classification. In *Advances in Neural Information Processing Systems*, 2018.

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*, pages 935–944, San Francisco, CA, 2016.

K. Jasinska and N. Karampatziakis. Log-time and Log-space Extreme Classification. *arXiv preprint arXiv:1611.01964*, 2016.

A. Kanehira, A. Shin, and T. Harada. True-negative label selection for large-scale multi-label learning. In *23rd nternational Conference on Pattern Recognition*, pages 3673–3678, Cancún Center, Cancún, México, 2016.

J.-J. Li, K. Lu, Z. Huang, and H.-T. Shen. Two birds one stone: On both cold-start and long-tail recommendation. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 898–906. ACM, 2017.

Y.F. Liang, C.-J. Hsieh, and Thomas C. M. Lee. Block-wise Partitioning for Extreme Multi-label Classification. 2018.

A. Niculescu-Mizil and M. E. Abbasnejad. Label filters for large scale multi-label classification. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1448–1457, Fort Lauderdale, FL, 2017.

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*, pages 263–272, New York, NY, 2014.

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. 2018.

S. Si, H. Zhang, S. S. Keerthi, D. Mahajan, I. S. Dhillon, and C.-J. Hsieh. Gradient boosted decision trees for high dimensional sparse output. In *Proceedings of International Conference on Machine Learning*, pages 3182–3190, 2017.

W. Siblini, P. Kuntz, and F. Meyer. CRAFTML, an efficient clustering-based random forest for extreme multi-label learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4664–4673, Stockholmsmässan, Stockholm Sweden, 2018.

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, 2017.

Y.-X. Wang and M. Hebert. Learning to learn: Model regression networks for easy small sample learning. In *Proceedings of European Conference on Computer Vision*, pages 616–634. Springer, 2016.

Y.-X. Wang, D. Ramanan, and M. Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pages 7032–7042, 2017.

J. Weston, A. Makadia, and H. Yee. Label partitioning for sublinear ranking. In *Proceedings of International Conference on Machine Learning*, pages 181–189, 2013.

C. Xu, D.-C. Tao, and C. Xu. Robust extreme multi-label learning. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1275–1284, San Francisco, CA, 2016.

L. Q. Yang, Y. Cui, Y. Xuan, C. Y. Wang, S. Belongie, and D. Estrin. Unbiased Offline Recommender Evaluation for Missing-Not-At-Random Implicit Feedback. In *RecSys*, Vancouver, Canada, 2018.

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 31st AAAI Conference on Artificial Intelligence*, pages 2838–2844, San Francisco, CA, 2017.

I. E.-H. Yen, X.-R. Huang, P. Ravikumar, K. Zhong, and I. S. Dhillon. Pd-sparse: A primal and dual sparse approach to extreme multiclass and multilabel classification. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 3069–3077, New York, NY, 2016.

H.-F. Yu, P. Jain, P. Kar, and I. S. Dhillon. Large-scale multi-label learning with missing labels. In *Proceedings of the 31st International Conference on Machine Learning*, pages 593–601, Beijing, China, 2014.

W.-J. Zhang, L.-W. Wang, J.-C. Yan, X.-F. Wang, and H.-Y. Zha. Deep extreme multi-label learning. *arXiv preprint arXiv:1704.03718*, 2017.