ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

A Simple and Effective Scheme for Data Pre-processing in Extreme Classification

Sujay Khandagale[1] and Rohit Babbar[2] *

1- Indian Institute of Technology Mandi, CS Department
Mandi, India

2- Aalto University, CS Department
Helsinki, Finland

Abstract. Extreme multi-label classification (**XMC**) refers to supervised multi-label learning involving hundreds of thousand or even millions of labels. It has been shown to be an effective framework for addressing crucial tasks such as recommendation, ranking and web-advertising. In this paper, we propose a method for effective and well-motivated data pre-processing scheme in XMC. We show that our proposed algorithm, PrunEX, can remove upto 90% data in the input which is redundant from a classification view-point. Our scheme is universal in the sense it is applicable to all known public datasets in the domain of XMC.

1 Introduction

Extreme Multi-label Classification (**XMC**) refers to supervised learning of a classifier which can automatically label an instance with a small subset of relevant labels from an extremely large set of all possible target labels. Machine learning problems consisting of hundreds of thousand labels are common in various domains such as product categorization for e-commerce [1, 2], hash-tag suggestion in social media [3], annotating web-scale encyclopedia [4], and image-classification. It has been demonstrated that, in addition to automatic labelling, the framework of XMC can be leveraged to effectively address learning problems arising in bid-phrase suggestion in web-advertising and recommendation systems [4]. In the scenarios of ad-display, by treating each query as label, automatic prediction of potential monetizable bid-phrases can be made in response to an advertisement. The growing significance of XMC in web-scale data mining is further highlighted by dedicated workshops in premier machine learning and data mining conferences (cf. workshops on extreme classification at NIPS 2015–2017 and WWW 2018 [5]).

The large number of labels in typical XMC problems entails that the number of training instances and the number of features is also large. For instance, the benckmark Delicious-200K data on the XMC repository [1] with approximately 200,000 labels consists of a similar scale of training instances in a sparse feature space of dimensionality 800,000. Each training instance is a sparse representation of a text document with tf-idf (term frequency - inverse document frequency) weights associated with each word in the document. Such large-scale category

*This work was done when Sujay was a student at Aalto University, Finland

[1] `http://manikvarma.org/downloads/XC/XMLRepository.html`

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

systems exhibit fit to power-law distribution of training instances among labels [6, 7, 8]. The large scale of the problem leads to substantially significant resources in terms of training time, and the parameter size of the learnt model.

In this work, we closely study the training data in XMC setting and find that there is a significant portion of training data which is redundant from the viewpoint of classification. Concretely, we propose `PrunEX`, and show that pruning the input by thresholding the tf-idf weights such that the all values below a certain threshold are removed, we can significantly reduce the training data size without losing the prediction power of the learning algorithm. The distribution of the weight for terms is shown for another Delicious-200K dataset in Figure 1, where it is demonstrated that 88% of the weights can be removed with only 0.1% decrease in the prediction accuracy.

The main idea for our intuition of removal of terms with very low tf-idf weight is similar to the process followed by a human annotator during the manual labelling task. The annotator who assigns the relevant labels parses the text by paying attention to the main terms in the document and ignoring the insignificant ones. Also, since `PrunEX` involves thresholding in the input space, this can be thought of as a dual to the idea of low magnitude removal of weights in the parameter space which is done post-training in `DiSMEC` [9]. Being a preprocessing step, the proposed method significantly reduces training data size, training time, and model size.

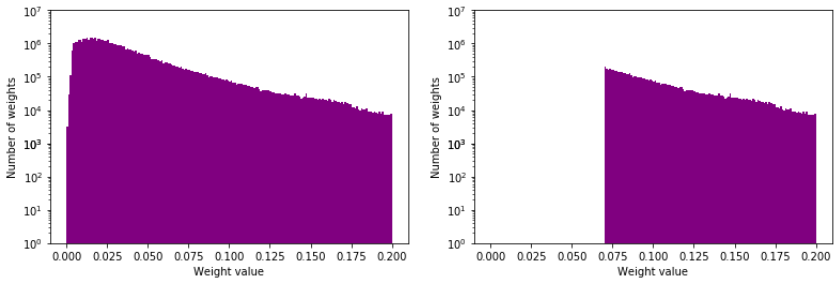

Fig. 1: Distribution of tf-idf weights before and after pruning for Delicious-200K dataset. The threshold was set to 0.07. The % of weights between 0.0 and 0.07 is 88%, weights larger than 0.2 are not shown for clarity.

1.1 Related Work

Most works on extreme multi-label classification fall in one of the three families : (i) label-embedding, (ii) One-vs-All, and (iii) tree-based approaches.

Label-embedding approaches These approaches [10, 11] usually make low-rank assumptions on the label space. However, a "global" low-rank assumption is usually violated in extreme classification setup with large number of tail-labels [9, 10]. Under this condition, embedding-based methods gives higher prediction error.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

One-vs-All approaches These approaches train one classifier per label, without making low-rank assumptions on the label space. Notable methods, such as `DiSMEC` [9], `ProXML` [12] and `PD-Sparse` [13] usually achieve better prediction performance, but require distributed infrastructure for efficient training. `DiSMEC` reduces model size by filtering out spurious model parameters and speeds up training and prediction by distributing the computation load to multiple machines.

Tree-based approaches These approaches [4, 14] aim towards faster training and prediction by recursively dividing the space of labels and/or features. However, due to the cascading effect in the tree structure, the prediction error made at a top-level cannot be corrected at lower levels. As a result, these methods have lower prediction accuracy. Typically, such techniques trade-off prediction accuracy for prediction speed which might be desired in some applications.

2 Proposed Algorithm: PrunEX

Let the training data be given by, $\mathbb{T} = \{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_N, \mathbf{y}_N)\}$ where $\mathbf{x}_i \in \mathbb{R}^D$ represents input feature vectors and $\mathbf{y}_i \in \{0,1\}^L$ represents their respective output vectors such that $y_{ij} = 1$ if the j-th label belongs to the training instance $\mathbf{x}_i$. Note that in XMC, the training set size N, the feature set dimensionality D and the label set size L are extremely large. Also, both the feature vectors and their corresponding output vectors are sparse i.e only a small subset of the feature space as well as the label space is active for a given training sample. However, given the scale of the datasets, the active subset of features for a training instance is still quite large.

In order to prune unimportant features, apprehending importance is essential. First the data is converted to term frequency-inverse document frequency (also called tf-idf) format. Term-Frequency (log normalized) of term j in a training instance i is given by $tf(j,i) = \log(1 + f_{j,i})$ where $f_{j,i}$ is the number of times j appears in instance i. Inverse Document Frequency measures how frequent is the term across instances, and it is given by $idf(j) = \log(1 + \frac{N}{n_j})$ where n_j is the number of times term j appears from the training set of size N. The tf-idf weight of each term is given by $\mathrm{x}_{ij} = tf(j,i) \times idf(j)$.

Each training instance $(\mathbf{x}_i, \mathbf{y}_i)$ is normalized such that the feature vector $\mathbf{x}_i$ to unit length as follows,

$$\hat{\mathrm{x}}_{i,j} = \frac{\mathrm{x}_{ij}}{\|\mathbf{x}_i\|}$$

Note that the feature normalization of one training instance is independent of the others. Also, in XMC, the features of a training instance are tf-idf values for the bag of words representation of that dataset. The tf-idf features represent the importance of words with respect to the whole dataset. However, the normalization of these tf-idf features per training instance further enhances their importance for their respective training instances.

Post normalizing the feature vector, `PrunEX` removes features from the normalized feature vector $\hat{\mathbf{x}}_i$ with feature values less than a certain threshold α.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Since, the feature vectors are sparse, unimportant features can simply be dropped from every training instance. This is given by

$$\hat{\mathrm{x}}_{i,j} = \begin{cases} \hat{\mathrm{x}}_{i,j} & \text{if } \hat{\mathrm{x}}_{i,j} > \lambda \\ 0 & \text{otherwise} \end{cases}$$

The choice of the threshold λ is fixed to 0.07 for most datasets. It may be noted that $0 \leq \hat{\mathrm{x}}_{i,j} \leq 1$ due to non-negativity and normalization constraints. Clearly, in this range 0.1 is a significant weight of a term in a document which may consist of potentially hundreds of words. Therefore, we choose a threshold 0.07 which was observed to be effective for most datasets.

Preprocessing the postprocessing In order to reduce the computational complexity as well as model size, the state-of-the-art `Parabel` and `DiSMEC` prune away spurious weights after training. However, this post-processing step doesn't reduce the computational cost while training the model. `PrunEX` preprocesses the training data by pruning away unimportant features from every training sample and thereby reduces the computational cost of training models.

The motivation for this thresholding based pruning comes from the way we humans classify documents. Whenever we are assigned the task to classify a document, we tend to focus on a small set of keywords in the document to pick the right tag for the given document. Many other words in the document do provide supporting information and context, however, they do not strongly influence our decision. A similar effect can be expected for learning algorithms.

3 Experimental Evaluation

Dataset	# Training	# Test	# Labels	# Features	APpL	ALpP
EURLex-4K	15,539	3,809	**3993**	5000	25.7	5.3
Wikipedia-31K	14,146	6,616	**30,938**	101,938	8.5	18.6
WikiLSHTC-325K	1,778,351	587,084	**325,056**	1,617,899	17.4	3.2
Wikipedia-500K	1,813,391	783,743	**501,070**	2,381,304	24.7	4.7
Amazon-670K	490,499	153,025	**670,091**	135909	3.9	5.4
Amazon-3M	1,717,899	742,507	**2,812,281**	337,067	31.6	36.1

Table 1: Multi-label datasets taken from the Extreme Classification Repository. APpL and ALpP represent average points per label and average labels per point respectively.

Dataset and evaluation metrics We perform empirical evaluation on publicly available datasets from the XMC repository curated from sources such as Amazon for recommendation tasks and Wikipedia for tagging tasks. The statistics of the datasets are shown in Table 1. The datasets exhibit a wide range of properties in terms of number of training instances, features and labels.

Given a label space of dimensionality L , a predicted label vector $\hat{\mathbf{y}} \in \mathbb{R}^L$ and a ground truth label vector $\mathbf{y} \in \{0,1\}^L$, we use standard evaluation metrics,

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Precision@k and normalized Discounted Cumulative Gain (nDCG@k) defined as:

$$Precision@k(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{k} \sum_{l \in rank_k(\hat{\mathbf{y}})} \mathbf{y}_l \quad (1)$$

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

where DCG@k $= \frac{\mathbf{y}_l}{\sum_{l=1} \frac{1}{\log(l+1)}}$, and $rank_k(\hat{\mathbf{y}})$ returns the k largest indices of $\hat{\mathbf{y}}$.

3.1 Experimental results

The comparison of `PrunEX + DiSMEC` against the base algorithm `DiSMEC` for precision@k and nDCG@k metrics is shown in Tables 2 and 3. In the last column, the tables also show the reduction in training data in percentage terms. It is clear that substantial reduction in the training data can be obtained with almost no loss in prediction accuracy. Furthermore, the reduction in training data leads to substantial decrease in training time of the base learning algorithm as well as model size compared to the original algorithm. For instance, it was observed that the training approximately gets halved after applying the pre-processing via `PrunEX`. Also, the same behavior was observed for across other tree-based algorithms such as `Parabel` and `FastXML`, which shows the universality of the proposed pre-processing scheme in the XMC regime.

Dataset	PrunEX + DiSMEC			DiSMEC			% features removed
	Prec@1	Prec@3	Prec@5	Prec@1	Prec@3	Prec@5	
EURLex-4K	82.46	70.01	58.46	82.85	70.37	58.69	63%
Wikipedia-31K	83.66	73.11	64.54	84.11	74.63	65.81	77.73%
Amazon-13K	93.24	78.64	63.67	93.4	79.1	64.1	51.22%
Amazon-670K	44.38	39.33	35.82	44.7	39.7	36.1	52.62%
WikiLSHTC-325K	62.50	41.41	30.76	64.4	42.5	31.5	30.72%
Delicious-200K	44.77	38.52	35.32	44.92	38.23	34.84	88.19%

Table 2: Comparison of Precision@k scores of `PrunEX + DiSMEC` and `DiSMEC`. The threshold α was set to 0.04 for smaller datasets Eur-Lex and Wiki-31K and to 0.07 for all the bigger datasets.

Dataset	PrunEx + DiSMEC			DiSMEC			% features removed
	nDCG@1	nDCG@3	nDCG@5	nDCG@1	nDCG@3	nDCG@5	
EURLex-4K	82.46	72.01	66.14	82.40	72.50	66.70	63%
Wikipedia-31K	83.66	73.92	65.02	85.20	74.60	65.90	77.73%
Amazon-13K	93.24	87.31	85.27	93.4	87.7	85.8	51.22%
Amazon-670K	44.38	41.72	40.18	44.7	42.10	40.50	52.62%
WikiLSHTC-325K	62.50	57.00	57.02	64.4	58.5	58.4	30.72%
Delicious-200K	44.77	40.00	37.54	45.50	40.90	37.80	88.19%

Table 3: Comparison of nDCG@k scores of `PrunEx + DiSMEC` and `DiSMEC`. The threshold α was set to 0.04 for smaller datasets Eur-Lex and Wiki-31K and to 0.07 for all the bigger datasets.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

4 Conclusion

In this paper, we proposed a method for effective and well-motivated data preprocessing scheme in extreme multi-label classification. The main idea is to retain only those tf-idf weights which are above a certain threshold, and zeroing those below it. This idea, which is motivated by how human annotators focus only on main terms while labelling, leads to drastic reduction in input training by removing redundant information, and yet retaining same level of prediction accuracy. Furthermore, the universality of the threshold across XMC datasets, and across algorithms demonstrates the effectiveness of our approach.

Acknowledgements

We appreciate the computing resources provided by the Aalto Science-IT project.

References

[1] Dan Shen, Jean David Ruvini, Manas Somaiya, and Neel Sundaresan. Item categorization in the e-commerce domain. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1921–1924. ACM, 2011.

[2] Samy Bengio, Jason Weston, and David Grangier. Label embedding trees for large multi-class tasks. In *Neural Information Processing Systems*, pages 163–171, 2010.

[3] Emily Denton, Jason Weston, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. User conditional hashtag prediction for images. In *KDD*, 2015.

[4] Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *KDD*, pages 263–272. ACM, 2014.

[5] Akshay Soni, Aasish Pappu, Robert Busa-Fekete, and Krzysztof Dembczynski. Extreme multilabel classification for social media chairs' welcome and organization. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 1893–1894, 2018.

[6] Rohit Babbar, Cornelia Metzig, Ioannis Partalas, Eric Gaussier, and Massih-Reza Amini. On power law distributions in large-scale taxonomies. *ACM SIGKDD Explorations Newsletter*, 16(1):47–56, 2014.

[7] Rohit Babbar, Ioannis Partalas, Eric Gaussier, and Massih R Amini. On flat versus hierarchical classification in large-scale taxonomies. In *Advances in Neural Information Processing Systems*, pages 1824–1832, 2013.

[8] Rohit Babbar, Ioannis Partalas, Eric Gaussier, and Massih-reza Amini. Re-ranking approach to classification in large-scale power-law distributed category systems. In *Proceedings of the 37th international ACM SIGIR*, pages 1059–1062. ACM, 2014.

[9] Rohit Babbar and Bernhard Schölkopf. Dismec: Distributed sparse machines for extreme multi-label classification. In *WSDM*, pages 721–729, 2017.

[10] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. Sparse local embeddings for extreme multi-label classification. In *NIPS*, 2015.

[11] Yukihiro Tagami. Annexml: Approximate nearest neighbor search for extreme multi-label classification. In *KDD*. ACM, 2017.

[12] Rohit Babbar and Bernhard Schölkopf. Adversarial extreme multi-label classification. *arXiv preprint arXiv:1803.01570*, 2018.

[13] Ian En-Hsu Yen, Xiangru Huang, Pradeep Ravikumar, Kai Zhong, and Inderjit Dhillon. Pd-sparse: A primal and dual sparse approach to extreme multiclass and multilabel classification. In *International Conference on Machine Learning*, pages 3069–3077, 2016.

[14] 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*, pages 993–1002, 2018.