Post-Processing Techniques for Improving Predictions of Multilabel Learning Approaches

Akshay Soni, Aasish Pappu, Jerry Chia-mau Ni* and **Troy Chevalier**
Yahoo! Research, USA
*Yahoo! Taiwan
akshaysoni, aasishkp, jerryni, troyc@oath.com

Abstract

In *Multilabel Learning* (MLL) each training instance is associated with a *set of labels* and the task is to learn a function that maps an unseen instance to its corresponding label set. In this paper, we present a suite of—MLL algorithm independent—post-processing techniques that utilize the conditional and directional label-dependences in order to make the predictions from any MLL approach more coherent and precise. We solve a constraint optimization problem over the output produced by any MLL approach and the result is a refined version of the input predicted label set. Using proposed techniques, we show absolute improvement of 3% on English News and 10% on Chinese E-commerce datasets for P@K metric.

1 Introduction

The *Multiclass Classification* problem deals with learning a function that maps an instance to its one (and only one) label from a set of possible labels while in MLL each training instance is associated with a *set of labels* and the task is to learn a function that maps an (unseen) instance to its corresponding label set. Recently, MLL has received a lot of attention because of modern applications where it is natural that instances are associated with more than one class simultaneously. For instance, MLL can be used to map news items to their corresponding topics in Yahoo News, blog posts to user generated tags in Tumblr, images to category tags in Flickr, movies to genres in Netflix, and in many other web-scale problems. Since all of the above mentioned applications are user-facing, a fast and precise mechanism for automatically labeling the instances with their multiple relevant tags is critical. This has resulted in the development of many large-scale MLL algorithms.

The most straightforward approach for MLL is *Binary Relevance* that treats each label as an independent binary classification task. This quickly becomes infeasible if either the feature dimension is large or the number of labels is huge or both. Modern approaches either reduce the label dimension, e.g., PLST, CPLST (Chen and Lin, 2012), Bayesian CS (Kapoor et al., 2012), LEML (Yu et al., 2014), RIPML (Soni and Mehdad, 2017), SLEEC (Bhatia et al., 2015), or feature dimension or both (such as WSABIE and DocTag2Vec (Chen et al., 2017)). The inference stage for all of these approaches produce a score for each potential label and then a set of top-scored labels is given as the prediction.

A potential problem with the above mentioned algorithms is that they lack the knowledge of correlation or dependency between the labels. Let us look at a toy example: our training data is such that whenever the label *mountain* is active then the label *tree* is also active. Therefore MLL algorithm should take advantage of this correlation embedded in the training data to always infer the label *tree* when *mountain* is one of the label. On the other hand, if *tree* is an active label then *mountain* may not be a label. Exploiting this directional and conditional dependency between the labels should allow us to predict a more coherent set of labels. It would—to some extend—also save the MLL algorithm from making wrong predictions since if some wrong labels (say we predict a wrong label *politics*) are predicted along with correct labels (when true labels are *tree* and *mountain*) then the overall set of predicted labels would not be coherent. Inclusion of this external knowledge about labels shows significant improvements when there is a lack of training data for some labels.

There have been many attempts (Dembszynski

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 61–66,
Taipei, Taiwan, November 27 – December 1, 2017

et al., 2010) of using label-hierarchies or label-correlation information as part of the MLL training process. For instance, the label-correlation in training data is used in (Tsoumakas et al., 2009); (Guo and Gu, 2011) uses conditional dependencies among labels via graphical models. Some of the other relevant works that use this information as part of the training are (Huang and Zhou, 2012; Kong et al., 2013; Younes et al., 2008) and references therein. Since these approaches use label dependency information as part of the training stage, we foresee following issues:

- Using pre-trained model: In some cases we want to use a pre-trained MLL model that did not use the label-dependency information during training and retraining a new model is not an option. The use of pre-trained models has become very common since not everyone has the hardware capability to train complex models using large amounts of data.
- Label-dependency information not available during training or else one may want to use updated or new label-dependency information after the model is trained.
- Expensive training and inference: Almost all algorithms that utilize the label-dependence as side-information are either expensive during training, or inference, or both.

In this paper, we present a suite of post-processing techniques that utilize the conditional and directional label-dependences in order to make the predictions from any MLL approach more coherent and precise. It is to be noted that the proposed techniques are algorithm independent and can even be applied over the predictions produced by approaches that use this or any other label-dependency information as part of the training. Our techniques involve solving simple constraint optimization problems over the outputs produced by any MLL approach and the result is a refined version of the input prediction by removing spurious labels and reordering the labels by utilizing the additional label-dependency side information. We show benefits of our approach on Chinese e-commerce and English news datasets.

2 Problem Description and Approaches

MLL is the problem of learning a function $f : \mathcal{I} \rightarrow 2^{\mathcal{L}}$ that maps an instance in $\mathcal{I}$ to one of the sets in the power set $2^{\mathcal{L}}$ where $\mathcal{L}$ is the set of all possible labels. For a specific instance, MLL predicts a subset $S \subset \mathcal{L}$. Our goal is to learn a subset $L \subset S$ such that L is a refined version of S.

Given a set of constraints on input labels, one can define an objective function that would potentially minimize inconsistencies between the final set of labels. Intuitively, labels may be interdependent, thus certain subsets are more coherent than the others. Label dependency can manifest either through human-curated label taxonomy or conditional probabilities. We propose two post-processing techniques in this paper to improve predicted outputs of any MLL algorithm. In the following subsections, we present details of each technique.

2.1 Steiner Tree Approximation

We formulate label coherence problem as a Steiner Tree Approximation problem (Gilbert and Pollak, 1968). Consider the following: input is a set of predicted labels $S = R \cup O$, where R is a set of coherent (required) labels and O is a set of incoherent (optional) labels. Labels are connected by directed weighted edges, thus form a graph G. The goal is to find a tree $T = (L, E, W)$ where L is a set of labels $R \subset L \subset S$ that includes all of the coherent labels and may include some of the optional labels O, E is the set of directed edges connecting nodes in L and W is set of weights assigned to the edges. For faster and approximate solutions, one can reduce Steiner tree problem to directed minimum spanning tree (MST) (Mehlhorn, 1988) and can be solved using Edmond's algorithm (Edmonds, 1967). MST has been applied in several previous works on document summarization (Varadarajan and Hristidis, 2006), text detection in natural images (Pan et al., 2011), and dependency parsing (McDonald et al., 2005). In this work, we first construct a directed graph of labels and then apply MST to obtain a tree of coherent labels. On applying MST, we choose vertices with top-K edge weights. Our goal is to find a tree that minimizes the following objective function:

$$\text{cost}_d(T) := \sum_{(u,v)\in E} d(u, v),$$

where u and v are nodes, $d(u, v) = 1 - W(u, v)$. The edge weights W are determined by the conditional probabilities of co-occurrence of labels. Directionality of the edges are determined by the

following criterion:

$$\begin{cases} L_i \rightarrow L_j, & \Pr(L_i|L_j) \leq \Pr(L_i|L_j) \\ L_i \leftarrow L_j, & \text{otherwise,} \end{cases}$$

where $\Pr(L_i|L_j)$ is the probability that label L_i is active given label L_j is active.

Once the directed graph is constructed based on above criterion, Edmond's algorithm recursively eliminates edges between a root node and one of the other nodes with minimum edge weights. In case of cycles, the edges are eliminated randomly. In essence, this algorithm selects highest-value connected-component in the directed graph. Thus, we are left with coherent labels.

2.2 0-1 Knapsack

Assigning labels to an instance with a budget can be considered as a resource allocation problem. `0-1` Knapsack is a popular resource allocation problem where capacity of the knapsack is limited and it can be filled with only a few valuable items. Items are added to the knapsack by maximizing the overall value of the knapsack subject to the combined weight of the items under budget. Many previous works in NLP have used Knapsack formulation, particularly in summarization (Lin and Bilmes, 2010; Ji et al., 2013; Hirao et al., 2013; McDonald, 2007). We formulate label assignment problem as a resource allocation problem, where we maximize total value of assigned labels. We determine individual value of a label based on the log likelihood of the label and its dependent labels. Intuitively, a label is included in the knapsack only when its dependent labels increase the overall log-likelihood.

$$\text{maximize} \quad \sum_{k \in S} \sum_{i \in D_k} \log(\Pr(L_k|L_i))$$
$$\text{s.t.} \quad \sum_{k \in S} |D_k| \leq C,$$

where $D_k \subset S - L_k$ is a subset of input labels S that are conditionally dependent on label L_k i.e. $\Pr(L_k|L_i) > 0$ for $i \in D_k$. To include a label L_k in the knapsack (i.e., in L), we optimize the total sum of the log conditional probabilities of labels L_k under the constraint that the total number of dependent labels are within the budget C—total number of permissible labels. The problem can be understood as a maximization of values of assigned labels. This problem is solved using a dynamic programming algorithm that runs in polynomial time (Andonov et al., 2000).

3 Experiments

The goal of this section is to emphasize on the fact that our post-processing techniques are MLL algorithm independent. For that we apply our approaches over the predictions from multiple MLL algorithms for two datasets: Yahoo News dataset in English and Chinese E-commerce dataset. Since MLL is generally used in applications where precision of predictions are important, we use Precision@K for $K = 1, 2$ and 3 as our metric.

3.1 Datasets

- **Yahoo News MLL Dataset (English)**[1]: This is one of the few publicly available large scale datasets for MLL. It contains 38968 Yahoo News articles in English for training and 10000 for testing. These are manually labeled with their corresponding category labels; overall, there are 413 possible labels.

- **Chinese E-commerce MLL dataset:** This is a propriety dataset that contains product descriptions of 230364 e-commerce products in Chinese for training and 49689 for testing. Each product is tagged with labels about the product categories; overall there are 240 tags.

3.2 MLL Approaches

Since our post-processing techniques are MLL algorithm independent, we picked three MLL approaches to apply our post-processing techniques: Naive Bayes, CNN, and DocTag2Vec. From our perspective, we can treat these approaches as black-box that for a given instance generate the set of predicted labels $S \subset \mathcal{L}$.

- **Naive Bayes (NB) MLL**: Given a sequence of words, the probability of a tag is evaluated by multiplying the prior probability of the tag and the probabilities of observing the words given the tag, pre-computed from the training data.

- **CNN MLL** (Kim, 2014): Originally designed for text classification tasks, the model views sequence of word embeddings as a matrix and applies two sequential operations: *convolution* and *max-pooling*. First, features

[1] available publicly via Webscope: https://webscope.sandbox.yahoo.com/catalog.php?datatype=l&did=84

Dataset	MLL Approach	P@K	Default	Highest Priors Baseline-1	Greedy Baseline-2	MST	Knapsack
Yahoo News	DocTag2Vec	1	0.6821	0.6277	0.5927	0.6942	**0.6976** (+1.5%)
		2	0.6461	0.6132	0.5836	**0.6689** (+2.2%)	0.6568
		3	0.6218	0.6052	0.5750	**0.6485** (+2.6%)	0.6203
Chinese Ecom	DocTag2Vec	1	0.5309	**0.5718** (+4.0%)	0.5563	0.5510	0.5331
		2	0.5454	**0.5748** (+2.9%)	0.5664	0.5716	0.5442
		3	0.4813	0.4928	0.5802	**0.5820** (+10%)	0.4884
Chinese Ecom	CNN	1	**0.8554**	0.7658	0.6898	0.8483	0.8479
		2	0.7387	0.7164	0.6545	**0.7814** (+4.2%)	0.7450
		3	0.6095	0.5921	0.6646	**0.7249** (+11.5%)	0.6287
Chinese Ecom	NB	1	0.8752	0.8526	0.7545	0.8982	**0.9057** (+3.0%)
		2	0.8481	0.8167	0.6738	0.8456	**0.8538** (+0.5%)
		3	0.7913	0.7519	0.7129	**0.8101** (+1.8%)	0.7385

Table 1: P@K for various values of K for the two datasets considered and for different MLL algorithms. Here default means not using a coherence stage. In brackets are shown the improvements in precision over default by the best performing coherence approach.

are extracted by a convolution layer with several filters of different window size. Then the model applies a max-over-time pooling operation over the extracted features. The features are then passed through a fully connected layer with dropout and sigmoid activations where each output node indicates the probability of a tag.

ail

- **DocTag2Vec** (Chen et al., 2017): Recently proposed DocTag2Vec embeds instances (documents in this case), tags, and words simultaneously into a low-dimensional space such that the documents and the tags associated with them are embedded close to each other. Inference is done via a SGD step to embed a new document, followed by k-nearest neighbors to find the closed tags in the space.

3.3 Post-Processing Techniques

- **Highest Priors** (Baseline-1): Given the training data, compute the prior probabilities of each label and re-rank labels in S according to the decreasing order of these prior probabilities to produce the new set L.

- **Greedy** (Baseline-2): Given the pairwise conditional probabilities among the output labels, select most probable pairs above certain threshold τ; we experimented with values in range `[0.01, 0.1]` and used $\tau = 0.06$ in the final experiments.

- **MST**: Steiner Tree Approximation via MST. The edge weights are computed via the conditional co-occurrence of the labels in the training data and the directionality is enforced via the criterion described in Section 2.1.

- **0-1 Knapsack**: We set $C = 15$ and solve the optimization problem described in Section 2.2.

3.4 Results

The P@K values are shown in Table 1 for the two datasets and for various coherency algorithms applied over multiple MLL approaches. The two baselines—highest priors and greedy—work reasonably well but the best performing approaches are MST and Knapsack. For most of the cases MST works well and even in the scenarios where Knapsack beats MST, they both are close in performance. By using a post-processing step for coherency we generally see a lift of around $2 - 4\%$ in most of the cases and sometimes a lift of more than 10% is observed. We note that one can design the problem with more deeper conditions i.e., $P(L_1|L_2, L_3 \dots L_k)$ but only single label dependency has been used in our experiments. With deeper dependencies, more training data is required to reliably learn prior probabilities. Also as the number of labels increase, the number of conditionals increases, thus the inference becomes computationally expensive.

Table 2: Example tags for various Yahoo News articles. Tags highlighted in red did not appear in true labels. Superscripts on the tags denote following D: Output from DocTag2Vec system (Default in Table 1), Knap: Output from Knapsack, MST: Output from Steiner Tree Approximation. Tags without superscript were not predicted at inference.

Doc 1	telecommunication^{D,MST},company-legal-&-law-matters^{D,Knap,MST}, mergers,-acquisitions-&-takeovers laws-and-regulations,entertainment^{D}, handheld-&-connected-devices^{D,MST}
Doc 2	fashion^{D,MST},clothes-&-apparel,hollywood^{D,MST} celebrity^{D,MST,Knap},entertainment^{D,MST,Knap},music^{D,MST},contests-&-giveaways^{D}
Doc 3	handheld-&-connected-devices^{D,MST,Knap},telecommunication^{D,MST,Knap},money investment-&-company-information,investment,sectors-&-industries^{D},internet-&-networking-technology^{D}
Doc 4	autos^{D,MST,Knap},strikes,financial-technical-analysis,company-earnings^{D,Knap}
Doc 5	public-transportation^{D,MST,Knap},travel-and-transportation^{D,MST,Knap} celebrity,music^{D,MST,Knap},transport-accident^{D,MST,Knap},entertainment^{D}
Doc 6	family-health^{D,MST,Knap},mental-health^{D,MST},biology,pregnancy^{D},parenting^{D,MST,Knap},tests-&-procedures^{D}
Doc 7	laws-and-regulations^{D,MST,Knap},company-legal-&-law-matters^{D,MST,Knap}, money,investment-&-company-information,investment,lighting-&-accessories^{D}

4 Discussion and Conclusion

Table 2 illustrates MLL output of sample documents from Yahoo News corpus. We observed Knapsack algorithm is more conservative at subset selection compared to MST. Tags predicted by *Default* system include tags that are related to true tags but do not appear in the true tag subset e.g., in `Doc 1` *handheld-&-connected-devices* is related to *telecommunications*, similarly `Doc 2` and `Doc 5` has one related tag and one spurious tag — in both cases MST and KNAPSACK prune the spurious tags. In `Doc 2` *music* is related/coherent and *contest-&-giveaways* is spurious/incoherent. In `Doc 5` *transport-accident* is related/coherent and *entertainment* is a spurious tag.

In this paper we presented two post-processing techniques to improve precision of any MLL algorithm. In addition to experiments discussed in the paper, we conducted experiments with other combinatorial optimization algorithms as used in previous works viz., facility location (p-median) (Alguliev et al., 2011; Ma and Wan, 2010; Cheung et al., 2009; Andrews and Ramakrishnan, 2008) and other graph-based centrality methods (Wolf and Gibson, 2004; Li et al., 2006; Guinaudeau and Strube, 2013). However, we did not observe significant improvement over default (unprocessed) output. While many approaches exist that utilize the label-correlation and dependency information during training, to the best of our knowledge, this is the first work that uses this knowledge as part of a post-processing step that is independent of MLL algorithms.

Acknowledgements

We are grateful to TC Liou, Chasel Su, Yu-Ting Chang, Brook Yang for their contributions to this project. We also wish to thank Kapil Thadani, Parikshit Shah and the anonymous reviewers for their feedback.

References

Rasim M Alguliev, Ramiz M Aliguliyev, and Chingiz A Mehdiyev. 2011. psum-sade: a modified p-median problem and self-adaptive differential evolution algorithm for text summarization. *Applied Computational Intelligence and Soft Computing* 2011:11.

Rumen Andonov, Vincent Poirriez, and Sanjay Rajopadhye. 2000. Unbounded knapsack problem: Dynamic programming revisited. *European Journal of Operational Research* 123(2):394–407.

Nicholas Andrews and Naren Ramakrishnan. 2008. Seeded discovery of base relations in large corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. ACL, pages 591–599.

Kush Bhatia, Himanshu Jain, Purushottam Kar, Prateek Jain, and Manik Varma. 2015. Locally non-linear embeddings for extreme multi-label learning. *CoRR* abs/1507.02743.

Sheng Chen, Akshay Soni, Aasish Pappu, and Yashar Mehdad. 2017. Doctag2vec: An embedding based multi-label learning approach for document tagging. In *2nd Workshop on Representation Learning for NLP*.

Yao-nan Chen and Hsuan-tien Lin. 2012. Feature-aware label space dimension reduction for multi-label classification. In *Advances in NIPS 25*, pages 1529–1537.

Jackie Chi Kit Cheung, Giuseppe Carenini, and Raymond T Ng. 2009. Optimization-based content selection for opinion summarization. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation*. ACL, pages 7–14.

Krzysztof Dembszynski, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. 2010. On label dependence in multilabel classification. In *ICML Workshop on Learning from Multi-label data*.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards B* 71(4):233–240.

EN Gilbert and HO Pollak. 1968. Steiner minimal trees. *SIAM Journal on Applied Mathematics* 16(1):1–29.

Camille Guinaudeau and Michael Strube. 2013. Graph-based local coherence modeling. In *ACL (1)*. pages 93–103.

Yuhong Guo and Suicheng Gu. 2011. Multi-label classification using conditional dependency networks. In *IJCAI*. volume 22, page 1300.

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-document summarization as a tree knapsack problem. In *EMNLP*. volume 13, pages 1515–1520.

Sheng-Jun Huang and Zhi-Hua Zhou. 2012. Multi-label learning by exploiting label correlations locally. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Heng Ji, Benoit Favre, Wen-Pin Lin, Dan Gillick, Dilek Hakkani-Tur, and Ralph Grishman. 2013. Open-domain multi-document summarization via information extraction: Challenges and prospects. In *Multi-source, Multilingual Information Extraction and Summarization*, Springer, pages 177–201.

Ashish Kapoor, Raajay Viswanathan, and Prateek Jain. 2012. Multilabel classification using bayesian compressed sensing pages 2645–2653.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* .

Xiangnan Kong, Michael K Ng, and Zhi-Hua Zhou. 2013. Transductive multilabel learning via label set propagation. *IEEE Transactions on Knowledge and Data Engineering* 25(3):704–719.

Wenjie Li, Mingli Wu, Qin Lu, Wei Xu, and Chunfa Yuan. 2006. Extractive summarization using inter- and intra-event relevance. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 369–376.

Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 NAACL*. ACL, pages 912–920.

Tengfei Ma and Xiaojun Wan. 2010. Multi-document summarization using minimum distortion. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, pages 354–363.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *European Conference on Information Retrieval*. Springer, pages 557–564.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings HLT and EMNLP*. ACL, pages 523–530.

Kurt Mehlhorn. 1988. A faster approximation algorithm for the steiner problem in graphs. *Information Processing Letters* 27(3):125–128.

Yi-Feng Pan, Xinwen Hou, and Cheng-Lin Liu. 2011. A hybrid approach to detect and localize texts in natural scene images. *IEEE Transactions on Image Processing* 20(3):800–813.

Akshay Soni and Yashar Mehdad. 2017. Ripml: A restricted isometry property based approach to multilabel learning. *arXiv preprint arXiv:1702.05181* .

Grigorios Tsoumakas, Anastasios Dimou, Eleftherios Spyromitros, Vasileios Mezaris, Ioannis Kompatsiaris, and Ioannis Vlahavas. 2009. Correlation-based pruning of stacked binary relevance models for multi-label learning.

Ramakrishna Varadarajan and Vagelis Hristidis. 2006. A system for query-specific document summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, pages 622–631.

Florian Wolf and Edward Gibson. 2004. Paragraph-, word-, and coherence-based approaches to sentence ranking: A comparison of algorithm and human performance. In *Proceedings of the 42nd ACL*. page 383.

Zoulficar Younes, Fahed Abdallah, and Thierry Denœux. 2008. Multi-label classification algorithm derived from k-nearest neighbor rule with label dependencies. In *Signal Processing Conference, 2008 16th European*. IEEE, pages 1–5.

Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit S. Dhillon. 2014. Large-scale Multi-label Learning with Missing Labels. In *ICML*.