A General Two-stage Multi-label Ranking Framework

Yanbing Xue
Department of Computer Science
University of Pittsburgh
yax14@pitt.edu

Milos Hauskrecht
Department of Computer Science
University of Pittsburgh
milos@pitt.edu

Abstract

In this paper we develop and study solutions for the multi-label ranking (MLR) problem. Briefly, the goal of multi-label ranking is not only to assign a set of relevant labels to a data instance but also to rank the labels according to their importance. To do so we propose a two-stage model that consists of: (1) a multi-label classification model that first selects an unordered set of labels for a data instance, and, (2) a label ordering model that orders the selected labels post-hoc in order of their importance. The advantage of such a model is that it can represent both the dependencies among labels, as well as, their importance. We evaluate the performance of our framework on both simulated and real-world datasets and show its improved performance compared to the existing multiple-label ranking solutions.

1 Introduction

Situations, where we describe observed objects using an ordered set of labels are quite common in our everyday life. Take, for example, a patient in the hospital. The patient's condition could be described by a list of diagnoses corresponding to various diseases the patient suffers from, with the most serious one being the first diagnosis listed by the human expert to describe the patient case.

Multi-label ranking (MLR) models, where the model assigns an ordered set of labels to data instances can be designed and learned in many different ways. A typical MLR model projects all possible labels one may assign to an instance into a real-valued space that reflects their rankings. However, such a model assumes individual label projections are independent, hence, it ignores the dependencies that may exist among labels assigned to the instance. This may lead to an inconsistent set of labels. In this work, we explore an alternative MLR model that relies on (1) a multi-label classification model that first selects an unordered set of labels for a data instance, and, (2) a label ranking model that orders the selected labels post-hoc. One advantage of such a model is that it can use a variety of existing multi-label classification models in its first step. Another advantage, is that the label ranking model (used in the second stage), orders only labels chosen by the first model, hence it can properly reflect various label dependencies incorporated into the first model. To translate the above idea into a working framework, we develop a new max-margin multi-label ranker to order post-hoc the output of an existing multi-label classifier.

We experiment with our new MLR framework on both synthetic and real-world datasets. We evaluate two aspects of our solution: (1) its ability to find the correct set of labels and (2) its ability to properly rank these labels. We show the effectiveness of our MLR framework by comparing its performance on both tasks with existing multi-label ranking solution.

2 Related Work

In this section, we briefly review literature in two areas related to our work: multi-label classification and multi-label ranking.

2.1 Multi-label Classification

In general, multi-label classification concentrates on learning a model that outputs a bipartite partition of all possible labels into relevant and irrelevant labels with respect to a data instance. Multi-label classification can be treated as an extension of multi-class classification: each instance is associated with a subset of labels instead of a single label. Multi-label classification can also be treated as an aggregation of multiple binary classification tasks with the same input features. The key to learning a multi-label classification model is the successful capture of the hidden dependencies among the labels.

Multiple methods have been proposed for learning a multi-label classification model. Perhaps the earliest and the most simple method is *binary relevance* [Boutell et al., 2004; Clare and King, 2001], which trains multiple binary classifiers independently. Clearly, the limitation of this method is that it totally ignores the dependencies among the labels. Another simple method is *label powerset* [Tsoumakas, Katakis, and Vlahavas, 2010], which transforms each label combination to a new class value, and learns a multi-class classifier with all the new class values. This method captures the dependencies among the labels by learning the full-joint of the labels. However, the limitation is also obvious: the number of new class values is exponential to the number of labels. Also, this method cannot learn the label combinations that are absent in the training data. There are also methods derived from these two simple methods:

max-margin output coding (MMOC) [Zhang and Schneider, 2012] applies a maximum margin formulation to encode the label combinations as new class values, and learns a multi-class classifier with all the new class values. The limitation of this method is the instability in performance: the effective capture of label dependencies is determined by the output coding algorithm. *Classification with heterogeneous features* (CHF) [Godbole and Sarawagi, 2004] first learns multiple binary classifiers independently, then trains multiple second-stage binary classifiers using the input features plus the output of the independent binary classifiers learned previously. The performance of this method is also unstable since the performance is highly dependent on the performance of the independent binary classifiers. More sophisticated multi-label classification methods based on probabilistic graphical models (PGMs) to exploit the conditional independence relations among the labels have been developed in recent years. *Conditional random field* (CRF) [Lafferty, McCallum, and Pereira, 2001; Naeini et al., 2014; Bradley and Guestrin, 2010] and *max-margin Markov network* (M3N) [Taskar, Guestrin, and Koller, 2003] use the feature vectors generated from the input features and the structure of the undirected graph (Markov network) to train a regression-based or a max-margin classification model respectively. *Conditional tree-structured Bayesian network* (CTBN) [Batal, Hong, and Hauskrecht, 2013] uses a directed acyclic graph to model the causal dependencies among labels and the input features; [Hong, Batal, and Hauskrecht, 2014, 2015] propose mixture frameworks to further improve the performance of CTBN. By modeling the conditional dependencies via Markov or Bayesian networks, probabilistic-graphical-model-based (PGM-based) methods can efficiently capture the hidden dependencies among labels and train models in polynomial time. Because of that, PGM-based methods for multi-label classification are gaining more and more popularity in recent years.

Although probabilistic-graphical-model-based (PGM-based) methods proved to be effective and efficient on multi-label classification tasks, it is infeasible to apply these methods to multi-label ranking tasks directly: the key and perhaps hardest challenge is the re-design of the loss functions. To avoid this, we propose a new general multi-label ranking framework that can be attached to most existing multi-label classifiers. It relies on a max-margin multi-label ranker that ranks only the relevant labels from the output of the existing multi-label classifier.

2.2 Multi-label Ranking (MLR)

Multi-label ranking (MLR) is a complex learning problem where the goal is to not only identify relevant labels from a set of predefined labels, but also to rank them according to their relevance to a data instance. Consequently, MLR can be considered as a generalization of multi-label classification and label ranking. The key to learning a successful MLR model is the capture of the dependencies among the labels.

Only a limited number of solutions have been developed for MLR problem. [Fürnkranz et al., 2008] proposed to add a threshold label and apply a common label ranking model using pairwise constraints: labels ranked beyond the threshold labels are positive (and are included among instance labels), otherwise they are negative (and not included among labels). Similarly, [Li, Song, and Luo, 2017] proposed a multi-label ranker using a smoothed hinge loss function to enforce the pairwise orderings between each pair of labels unless both labels are irrelevant. [Jung and Tewari, 2018] proposed an online boosting algorithm to rank the labels by combining the predictions of multiple weak regression-based models. Both of these methods come with the following drawbacks: (1) they do not explicitly learn the dependencies among labels, hence they fail to capture many label dependencies, such as situations in which two labels are mutually exclusive; (2) they take marginalized predictions to determine the label rankings and the relevance of the labels. Such an approach is is unable to properly model the dependencies among the labels and their rankings.

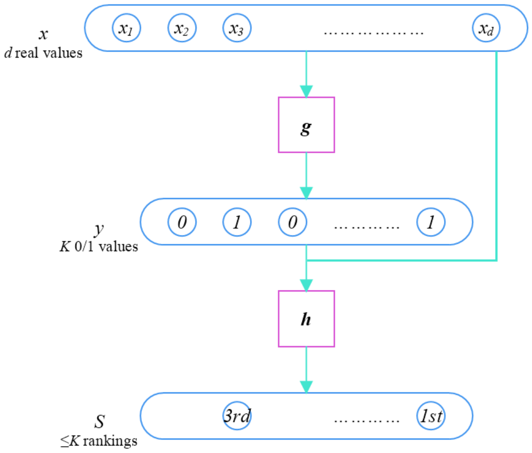

Figure 1: A two-stage MLR model **f** consisting of a multi-label classifier **g** and an auxiliary multi-label ranker **h**. The number of labels in permutation subset S equals the number of positive labels in label vector **y**.

To solve the drawbacks of the existing MLR methods, we propose a general MLR framework that can be combined with many existing multi-label classifiers to both (1) capture the dependencies among the labels (2) rank the relevant labels from the output of the existing multi-label classifier.

3 Methodology

In this section, we start by first defining the problem of learning a multi-label ranking (MLR) model and propose a simple two-stage model for the problem. The model consists of a multi-label classifier and a new label ranker model and their composition. Since there are many different multi-label classification we focus on and present a new multi-label ranker model responsible for ordering the labels selected by the existing multi-label classifier.

3.1 Problem

Our objective is to learn an MLR model $\mathbf{f} : X \rightarrow \mathbf{S}$, where $X \in \mathbb{R}^d$ is the input space and $\mathbf{S}$ represents the space of the permutation subsets (PS). The PS $S^{(i)}$ reflects the rank-

Dataset	Instance Number	Feature Number	Label Number	Set Number	Cardinality
Emotions	593	72	6	27	1.9
Yeast	2417	103	14	198	4.2
Scene	2407	294	6	15	1.1
MS1	35409	90	13	156	1.3
MS2	89073	90	15	210	1.3
Faces	584	256	4	23	1.4

Table 1: Properties of all datasets in experiments.

ings of the relevant labels in terms of their importance to the instance among all the K labels. The PS $S^{(i)}$ is formed by a non-empty subset of K labels indicating the descending ordering of the relevant labels. The labels not in the PS are considered irrelevant to the instance by the annotator. For example, in a 4-label setting, a PS $\langle 3, 2\rangle$ indicates the 3rd label is the most relevant to the instance, the 2nd label is the second most relevant, and the other two labels are irrelevant.

3.2 The Model

The model of $\mathbf{f}$ that assigns a set of ordered labels to instances can be built in many different ways. In this work we adopt a two-step process covered with two different models to define it: $\mathbf{f} = \langle \mathbf{g}, \mathbf{h}\rangle$. The first model is a multi-label classifier $\mathbf{g} : X \to Y$ where $Y = \{0, 1\}^K$ is the space of the label vector. Such a classifier determines whether a specific label $y_j^{(i)}$ in the label vector $\mathbf{y}^{(i)}$ is relevant to the instance $\mathbf{x}^{(i)}$ or not ($y_j^{(i)} = 1$ indicates relevant). The second model is an multi-label ranker $\mathbf{h} : X \times Y \to \mathbf{S}$ that determines the ordering of the relevant labels in $\mathbf{y}^{(i)}$ and outputs it as $S^{(i)}$. A brief illustration of this MLR model $\mathbf{f}$ is in **Figure 1**.

We note that a large body of research work in recent years has focused on the multi-label classification problem, and many different multi-label classification models have been proposed and developed. Our goal in this work is not to invent a new multi-label classification model, but to utilize the existing models in our two-step MLR model.

3.3 An Auxiliary Max-margin Multi-label Ranker

Suppose we have access to a multi-label classifier $\mathbf{g}$ that outputs a label vector $\mathbf{y}^{(i)}$ which determines whether a label is relevant to the instance (in the PS) or not. Then we can train an auxiliary max-margin multi-label ranker $\mathbf{h}$ on the label vectors such that, for each instance, the projection of a label in the PS should be higher than all other labels that rank lower in the PS. More formally, suppose that we have already obtained a label vector $\mathbf{y}^{(i)}$ where $y_j^{(i)} = 1$ indicates label j is included in the PS. Now, we aim to obtain K different projection mappings $h_1, h_2, \ldots, h_K$, one for each label, that reflect their order in the PS $S^{(i)}$. We can encode this aim by trying to enforce the following constraints: $h_j(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) > h_l(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \Leftrightarrow r(S^{(i)}, j) < r(S^{(i)}, l)$, that is, the projection h_j of label j should be higher than the projection h_l of any label l such that the ranking $r(S^{(i)}, j)$ of label j in $S^{(i)}$ is beyond the ranking $r(S^{(i)}, l)$ of label l. Particularly, if $j \notin S^{(i)}$, $r(S^{(i)}, j) = |S^{(i)}| + 1$. Therefore, our auxiliary max-margin multi-label ranker can be formulated as the following optimization problem:

$$\min_{W,\Xi} \sum_{j=1}^{K} R(\mathbf{w}_j) + C \sum_{i=1}^{N} \sum_{j=1}^{K-1} \sum_{l=j+1}^{K} \xi_{jl}^{(i)}$$

$$z_{jl}^{(i)} (\mathbf{w}_j - \mathbf{w}_l)^\top \phi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \geq 1 - \xi_{jl}^{(i)} \qquad (\xi_{jl}^{(i)} \geq 0)$$

where $\mathbf{w}_j \subset W$ is the model parameter of h_j; $R(\mathbf{w}_j)$ is the regularization term of h_j; $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ are the feature and label vector of instance i obtained from the given multi-label classifier $\mathbf{g}$, respectively; $\phi(\cdot)$ is the projection of kernel space; $z_{j,l}^{(i)}$ is the ternary value indicating the comparison of the rankings between label j and l: 1 if $r(S^{(i)}, j) < r(S^{(i)}, l)$, and -1 if $r(S^{(i)}, j) > r(S^{(i)}, l)$, and 0 otherwise; $\xi_{j,l}^{(i)} \in \Xi$ is the slack variable penalizing when the comparison between $h_j(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ and $h_l(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ violates their rankings in $S^{(i)}$; N is the number of labeled instances. Clearly, the total number of constraints of our auxiliary max-margin multi-label ranker is $O(Nv^2)$, where v is the average size of the PS of each instance.

4 Experiments and Results

We test our model and learning solutions on multiple synthetic and real-world datasets. The three synthetic datasets are built from UCI multi-label classification datasets where the permutation subsets are simulated; the three real-world datasets contain permutation subsets provided by human annotators.

4.1 Datasets

The synthetic datasets are generated from UCI multi-label classification datasets. We generate them by taking $\frac{1}{3}$ of data instances randomly to train an identical multi-label ranking model with $0/1$ label vectors only. This is possible since we can still enforce that the projections of relevant labels should be higher than the projections of irrelevant labels. After training, we apply the trained multi-label ranking model to every instance in the remaining $\frac{2}{3}$ of the dataset and calculate the rankings of all its labels. By combining the label vector and the predicted rankings, we generate permutation subsets for every instance in the remaining $\frac{2}{3}$ of the dataset. In the experiments, we use only the data instances randomly sampled from $\frac{2}{3}$ of data that consists of the original feature vectors and the generated permutation subsets.

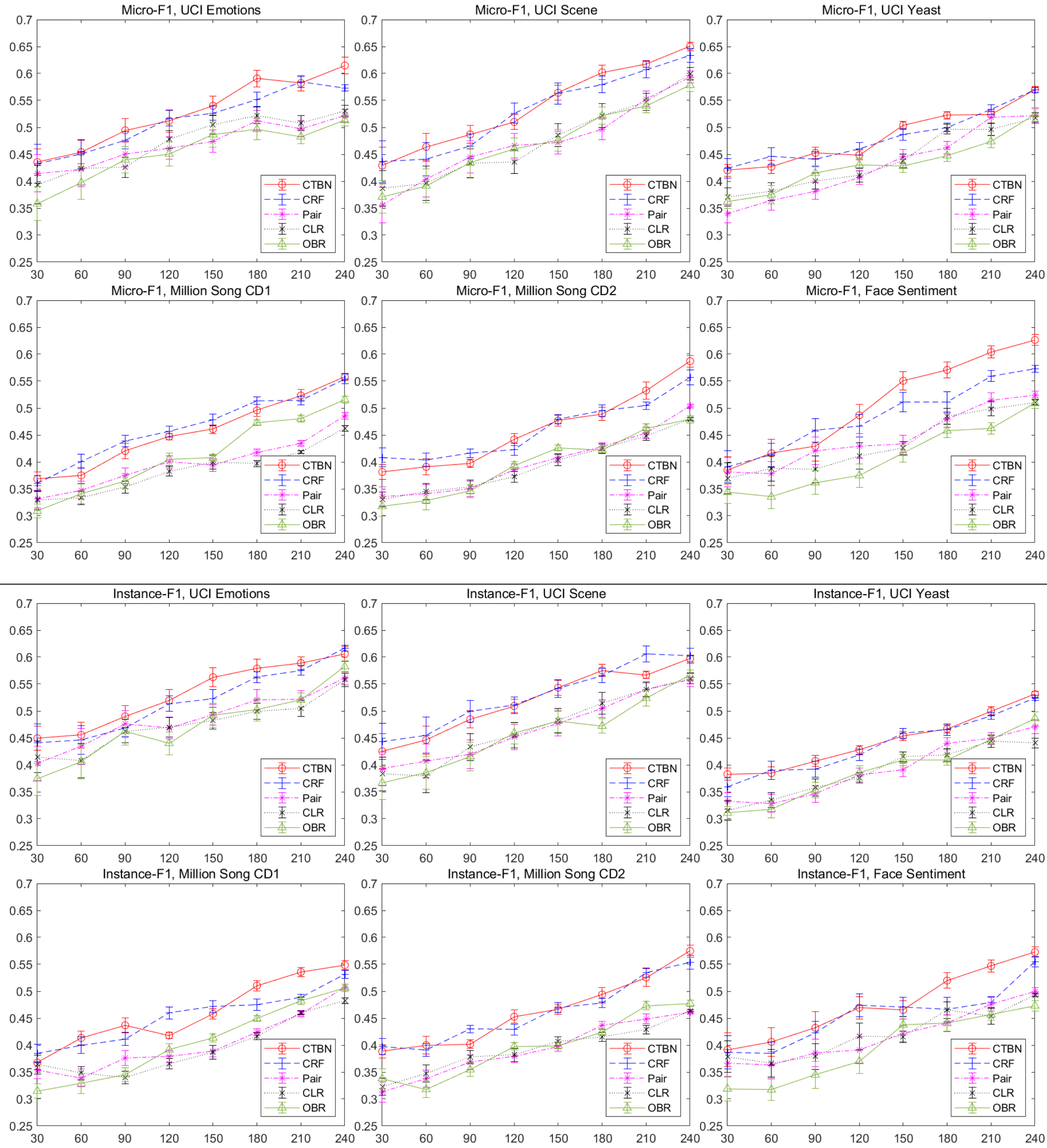

Figure 2: Classification Performance (Top – Micro-F1, Bottom – Instance-F1) on all datasets.

The real-world datasets consists of two Million Song datasets (MS1 and MS2) [Bertin-Mahieux et al., 2011] and one Face Sentiment dataset [Mozafari et al., 2012]. Each Million Song dataset consists of a collection of songs. In each dataset, the feature vector of an instance (song) contains the timbre information of the song, and the permutation subset of each instance contains one or two labels indicating the priorities of the genres. In Face Sentiment data, the feature of each instance is a 128×120 gray-scale image of a facial expression, where we extract 256 features using a multilayer convolutional neural network. The output of each instance indicates one sentiment of facial expression out of four provided by nine human annotators. Therefore, we may sort the output sentiment according to their vote numbers in the descending order, and take such an ordered set as the permutation assigned to each instance. The properties of six datasets are summarized in **Table 1**.

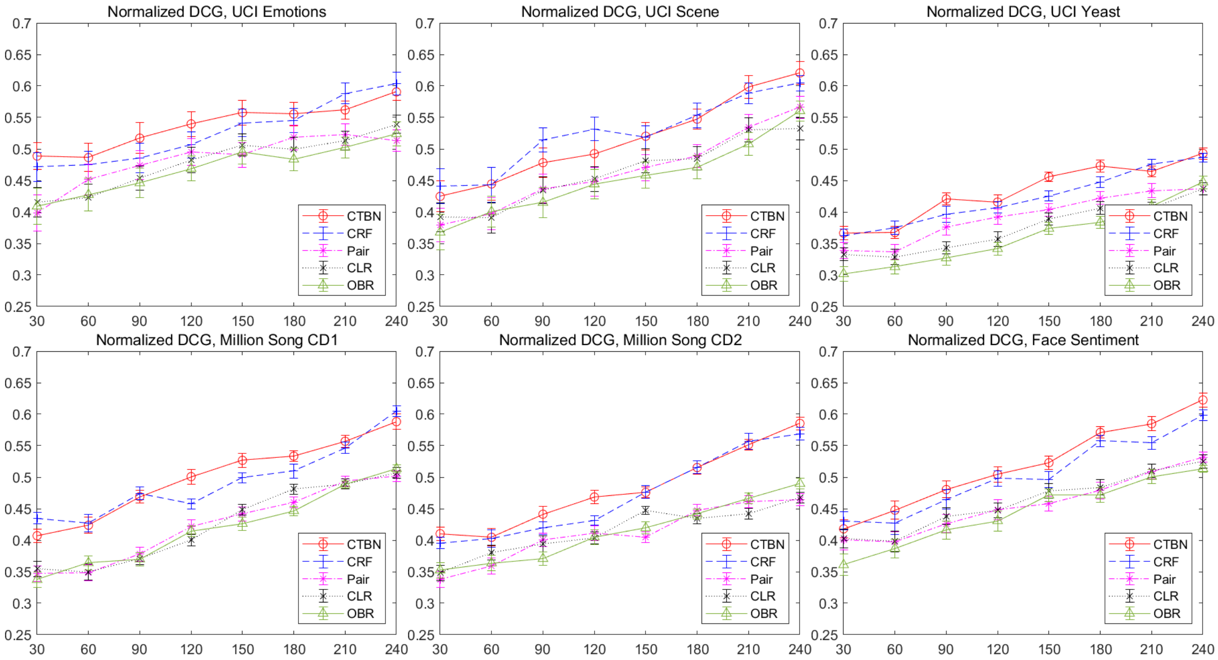

Figure 3: Ranking Performance (Normalized Discounted Cumulative Gain) on all datasets.

4.2 Settings

To demonstrate the benefits of our model we evaluate and compare the performance of the following five models:

CTBN, a combination of the conditional tree-structured Bayesian network (CTBN) [Batal, Hong, and Hauskrecht, 2013] classifier and our multi-label ranker. CTBN models the conditional dependencies among labels via a Bayesian network;

CRF, a combination of the conditional random field (CRF) [Lafferty, McCallum, and Pereira, 2001; Bradley and Guestrin, 2010] classifier and our multi-label ranker. CRF models the conditional dependencies among labels via a Markov network;

Pair, a multi-label ranker proposed by [Li, Song, and Luo, 2017] that uses a smoothed hinge loss function to combine the constraints from pairwise ordering extracted from the label rankings in the permutation subset of each instance. This method does not explicitly model the dependencies among labels;

CLR, the calibrated labeled ranker proposed by [Fürnkranz et al., 2008] that adds a threshold label of each instance and apply a common label ranker using pairwise constraints. This method does not explicitly models the dependencies among labels;

OBR, the online boosting multi-label ranking model proposed by [Jung and Tewari, 2018] that aggregates the predictions of multiple weak multi-label rankers via majority votes. This method does not explicitly models the dependencies among labels.

Briefly, *CTBN* and *CRF* are two versions of our MLR framework that take advantage of two existing multi-label classification models. *Pair*, *CLR* and *OBR* are three existing multi-label ranking solutions.

All data sets are split into the training and test set (using $\frac{2}{3}$ and $\frac{1}{3}$ of all instances respectively). We evaluate the performance of all methods on the test data using both multi-label classification and ranking performance measures. Micro-F1 and Instance-F1 are two multi-label classification evaluation metrics that only consider the labels picked up by the models, not their order. Micro-F1 is the F-measure averaging over the prediction matrix and is defined as:

$$\text{Micro-F1}(Y, \hat{Y}) = \frac{2\sum_{i=1}^{N}\sum_{j=1}^{m} y_j^{(i)} \hat{y}_j^{(i)}}{\sum_{i=1}^{N}\sum_{j=1}^{m} y_j^{(i)} + \sum_{i=1}^{N}\sum_{j=1}^{m} \hat{y}_j^{(i)}}$$

where $y_j^{(i)} \in Y$ is the ground truth of the jth binary label for the ith instance; $\hat{y}_j^{(i)} \in \hat{Y}$ is the prediction of the jth binary label for the ith instance; N is the instance number; m is the label number. Instance-F1 is the F-measure averaging over all instances and is defined as:

$$\text{Instance-F1}(Y, \hat{Y}) = \frac{1}{N}\sum_{i=1}^{N} \frac{2\sum_{j=1}^{m} y_j^{(i)} \hat{y}_j^{(i)}}{\sum_{j=1}^{m} y_j^{(i)} + \sum_{j=1}^{m} \hat{y}_j^{(i)}}$$

Normalized Discounted Cumulative Gain (NDCG) [Järvelin and Kekäläinen, 2002; Li, Burges, and Wu, 2007] is used to evaluate the ranking performance. This evaluation metric considers the rankings of the relevant (positive) labels and is defined as:

$$\text{NDCG}(S, \hat{S}) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{Z^{(i)}} \sum_{j=1}^{m} I(S^{(i)}, j) DW(\hat{S}^{(i)}, j)$$

where $S^{(i)} \subset S$ is the ground truth of the PS for the ith instance indicating the rankings of the relevant (positive) labels; $\hat{S}^{(i)} \subset \hat{S}$ is the predicted PS for the ith instance; $I(S^{(i)}, j)$ returns the importance of the jth label in the ground-truth PS $S^{(i)}$. If the $j \in S^{(i)}$, indicating $y_j^{(i)} = 1$, then $I(S^{(i)}, j) = |S^{(i)}| + 1 - r(\hat{S}^{(i)}, j)$

where $r(\hat{S}^{(i)}, j)$ return the ranking of the jth label in $S^{(i)}$; if the $j \notin S^{(i)}$, indicating $y_j^{(i)} = 0$, then $I(S^{(i)}, j) = 0$. $DW(\hat{S}^{(i)}, j)$ returns the discounted weight of the jth label in the predicted PS $\hat{S}^{(i)}$. If the $j \in \hat{S}^{(i)}$, then $DW(\hat{S}^{(i)}, j) = \frac{1}{\log_2(1+r(\hat{S}^{(i)},j))}$, otherwise $DW(\hat{S}^{(i)}, j) = 0$. $Z^{(i)} = \sum_{j=1}^{m} I(S^{(i)}, j) DW(S^{(i)}, j)$ is the partition function, indicating the ideal discounted cumulative gain when $S^{(i)} = \hat{S}^{(i)}$.

All three evaluation metrics are measured on the test data regarding different numbers of labeled instances. The learning considers the training data only, and the three metrics are always calculated on the test set. We also repeat the train/test splitting and learning steps 30 times. The average classification performance (Y-axis) of different models for increasing sizes (X-axis) of the training sets is reported in **Figure 2**. The average ranking performance (Y-axis) of different models for increasing sizes (X-axis) of the training sets is reported in **Figure 3**.

4.3 Experimental Results

Results on Classification Performance In **Figure 2** we show the classification performance (Micro-F1 and Instance-F1) of the different multi-label ranking solutions. The two classification performance measures only compare the label vectors with the ground truth, not the order of labels. Both *CTBN* and *CRF* (our MLR solutions) outperform three existing MLR solutions *Pair*, *CLR* and *OBR* in terms of Micro-F1 and Instance-F1 on all the six datasets. This shows the effectiveness of the multi-label classifiers based on probabilistic graphical models (PGMs) we have integrated into our MLR framework. By modeling dependencies among labels, *CTBN* and *CRF* can improve the classification performance over existing multi-label rankers that do not explicitly model the dependencies among labels.

Results on Ranking Performance In **Figure 3** we show the ranking performance (Normalized Discounted Cumulative Gain) of different multi-label ranking frameworks. The ranking performance compares both the label vectors and the rankings of the relevant (positive) labels with the ground truth. Similarly to label classification performance, *CTBN* and *CRF* outperform *Pair*, *CLR* and *OBR* also in the ranking performance. This shows our two-stage multi-label ranker can effective utilize the label dependency learned from its stage-one multi-label classifier. The classification and ranking performance together shows the effectiveness of combining the existing multi-label classifiers based on probabilistic graphical models (PGMs) with our auxiliary multi-label ranker by modeling and utilizing the dependencies among labels.

5 Conclusion

We have proposed a new multi-label ranker that relies on outputs of existing multi-label classifiers, which can represent both the dependencies among labels as well as their importance. Through extensive experiments we showed that our new two-stage MLR approach can assign better label rankings to instance than existing state-of-the-art label ranking solutions.

References

Batal, I.; Hong, C.; and Hauskrecht, M. 2013. An efficient probabilistic framework for multi-dimensional classification. 2417–2422.

Bertin-Mahieux, T.; Ellis, D. P.; Whitman, B.; and Lamere, P. 2011. The million song dataset. In *Proc. of the 12th Int. Conf. on Music Inf. Retrieval (ISMIR 2011)*.

Boutell, M.; Luo, J.; Shen, X.; and Brown, C. 2004. Learning multi-label scene classification. *Pattern Recognition* 37:1757–1771.

Bradley, J. K., and Guestrin, C. 2010. Learning tree conditional random fields. In *Proc. of the 27th Int. Conf. on Int. Conf. on Machine Learning*, ICML'10, 127–134.

Clare, A., and King, R. D. 2001. Knowledge discovery in multi-label phenotype data. In *Principles of Data Mining and Knowledge Discovery*, 42–53.

Fürnkranz, J.; Hüllermeier, E.; Loza Mencía, E.; and Brinker, K. 2008. Multilabel classification via calibrated label ranking. *Machine Learning* 73(2):133–153.

Godbole, S., and Sarawagi, S. 2004. Discriminative methods for multi-labeled classification. In *Advances in Knowledge Discovery and Data Mining*, 22–30.

Hong, C.; Batal, I.; and Hauskrecht, M. 2014. A mixtures-of-trees framework for multi-label classification. In *Proc. of the 23rd ACM Int. Conf. on Conf. on Inf. and Knowledge Management*, CIKM '14, 211C220.

Hong, C.; Batal, I.; and Hauskrecht, M. 2015. A generalized mixture framework for multi-label classification. *Proc. of the SIAM Int. Conf. on Data Mining* 2015:712–720.

Järvelin, K., and Kekäläinen, J. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.* 20(4):422–446.

Jung, Y., and Tewari, A. 2018. Online boosting algorithms for multi-label ranking. In *Proc. of the 21st Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the Eighteenth Int. Conf. on Machine Learning*, ICML '01, 282–289.

Li, P.; Burges, C. J. C.; and Wu, Q. 2007. Mcrank: Learning to rank using multiple classification and gradient boosting. *Advances in Neural Inf. Processing Systems*.

Li, Y.; Song, Y.; and Luo, J. 2017. Improving pairwise ranking for multi-label image classification. In *The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Mozafari, B.; Sarkar, P.; Franklin, M. J.; Jordan, M. I.; and Madden, S. 2012. Active learning for crowd-sourced databases. *CoRR* abs/1209.3686.

Naeini, M. P.; Batal, I.; Liu, Z.; Hong, C.; and Hauskrecht, M. 2014. An optimization-based framework to learn conditional random fields for multi-label classification. In *Proc. of the 2014 SIAM Int. Conf. on Data Mining*, 992–1000. SIAM.

Taskar, B.; Guestrin, C.; and Koller, D. 2003. Max-margin markov networks. In *Proc. of the 16th Int. Conf. on Neural Inf. Processing Systems*, NIPS'03, 25–32.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2010. *Mining Multi-label Data*. 667–685.

Zhang, Y., and Schneider, J. 2012. Maximum margin output coding. In *Proc. of the 29th Int. Conf. on Machine Learning (ICML 2012)*.