Label Ranking with Semi-Supervised Learning

Wei Wang, Irwin King

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, New Territories, Hong Kong
{wangwei, king}@cse.cuhk.edu.hk

Abstract. Label ranking is considered as an efficient approach for object recognition, document classification, recommendation task, which has been widely studied in recent years. It aims to learn a mapping from instances to a ranking list over a finite set of predefined labels. Traditional solutions for label rankings cannot obtain satisfactory results by only utilizing labeled data and ignore large amount of unlabeled data. This paper introduces a novel Semi-Supervised Learning (SSL) framework by exploiting unlabeled data to improve the performance. Under this framework, we also propose a new Information Gain Decision Tree(IGDT) with aims to make full use of latent information and as such raise the efficiency and accuracy. Then we outline our models involving another two algorithms, Instance Based Learning (IBL) and Mallows Model Decision Tree (MMDT) within this framework. Experiment results demonstrate our approaches can obtain a better performance comparing with only applying labeled data.

Key words: Semi-Supervised Learning, label ranking

1 Introduction

Label ranking [4] [12] is a complex predication task aims to learn a mapping from instances to a list of rankings over a finite set of predefined labels. It can be viewed as a natural generalization of traditional classification with the reason that once obtaining the list of rankings, the first label is the result of single-label classification, and by setting a proper threshold of all labels, we can also get the result of multi-label classification [2]. A good application of label ranking is recommendation system, where users share the similar characteristics (e.g.: gender, age, occupation) will largely has the similar interests upon different movies (labels). The job of this learning algorithm is to assign higher rankings to the more relevant movies.

Many approaches for label ranking have been proposed recently. Constraint classification learns a framework capturing many flavors of multi-class classification including multi-label classification and ranking, and present a meta-algorithm for learning in this framework [6]. Log-linear models for label ranking [4] assumes that each instance in the training data is associated with a list of preferences over the label-set and learn a ranking function that induces a total order over the entire set of labels. As to the ranking by pairwise comparison, a binary preference model is learned for each pair of labels [7].

Even though those approaches have strong theoretical supports, they are limited to the supervised learning paradigm. With enough labeled data, they have shown good performance. However, in the real-world label ranking, unlabeled data are widely available. Another practical problem in label ranking is the distinctiveness of unlabeled data. It is difficult to assign a complete ranking to each instance, and as such plenty data are incomplete. Like the movie recommendation system, a common situation is that one user prefers action movie to comedy, but without any information about the science fiction film and thriller. Here action movie, comedy, science fiction and thriller are different labels in the system.

In this paper, we focus on another view to solve label ranking problem under the semi-supervised learning (SSL) framework. SSL [8], as a popular research field, provides a framework to solve the classification problem, including EM algorithm [5], graph-based methods [13], co-training methods [1]. The advantages of SSL is that by combining few labeled data and a large amount of unlabeled data (latent information), it can predict labels for unlabeled data. In this particular classification problem of label ranking, unlabeled data can be divided into unlabeled ranking and partial ranking. Unlabeled ranking means that, all labels of the corresponding ranking list are hidden, which is similar with the traditional unlabeled data. Partial ranking means only some labels are obtainable of the ranking list, while some are unobtainable. In order to utilize the available information as much as possible, we solve the label ranking problem by three methods under the framework of SSL.

In summary, our main contributions include:

1. Semi-supervised learning framework is firstly used for label ranking problem.
2. Based on the traditional information gain decision tree, we propose a new method in terms of the large amount of unlabeled data in label ranking problem.

In Section 2, we define the problem setup and introduce the SSL framework. Besides, we detail how three SSL methods, information gain decision tree, instance based label ranking [3], Mallows model decision tree [3], are applied to label ranking. In Section 3, we report the experimental comparisons and results. We conclude the paper in Section 4.

2 Label Ranking and the framework of SSL

2.1 Notation and Problem Setup

Label ranking is the task of inferring a total order over a predefined set of labels for each unlabeled instance. We denote the instance space as X and the label space as $L = \{l_i\}_{i=1}^{n}$, where $l_i \in \{1, ..., n\}$. n denotes the number of classes according to this data set.

Label Ranking Given a set of instances $X = \{x_i\}_{i=1}^{l+u+p}$, where $x_i \in R^m$, and whose corresponding ranking is $Y = \{y_i\}_{i=1}^{l+u+p}$. Each ranking y_i stands for a permutation of all labels from L. We use $y_i(j)$ to denote one single label, where $j \in \{1, ..., n\}$, and $\succ$ to denote the preference of different labels. n is the number of classes and $l + u + p$ is the number of instances.

According to this definition, $y_i(1) \succ y_i(2)$ expresses $y_i(1)$ is preferred to $y_i(2)$. Label ranking problem aims to learn an order of rankings in the form of $X \rightarrow Y$ mapping. We denote the resulting permutation $y_i = (y_i(1), ..., y_i(n))$. Here y_i can be viewed as a function from the identity permutation to another permutation.

Labeled Data We define the labeled data as $X = \{x_i\}_{i=1}^{l}$, where $x_i \in R^m$, with the corresponding ranking is $Y = \{y_i\}_{i=1}^{l}$, where y_i, $i \in \{1, 2, \dots l\}$ is an all permutation of Y from 1 to n.

Unlabeled Data We define the unlabeled data as $X = \{x_i\}_{i=l+1}^{l+u}$, where $x_i \in R^m$, with the corresponding ranking is $Y = \{y_i\}_{i=l+1}^{l+u}$. However, the label ranking information y_i is missing or latent, where $i \in \{l + 1, l + 2, \dots l + u\}$. u is the number of unlabeled data.

Partial Data We define the partial data as $X = \{x_i\}_{i=l+u+1}^{l+u+p}$, where $x_i \in R^m$, with the corresponding ranking is $Y = \{y_i\}_{i=1+u+1}^{l+u+p}$. Each y_i is a subset of all permutation from 1 to n, $i \in \{l+u+1, l+u+2, \dots l+u+p\}$. p is the number of partial data.

To evaluate the predictive performance of the mapping function, a suitable loss function on Y is necessary. Diverse methods are used to calculate this distance, here we select a popular one in statistics, which is called Kendall tau rank coefficient [11] [9]. Suppose y and z are two rankings, we define the distance between them as follows:

$$D(y, z) = \#\{(i, j) | (y(i) - y(j))(z(i) - z(j)) < 0\}. \quad (1)$$

$D(y, z)$ will be equal to 0 if the two lists are the same and n(n - 1) / 2 (where n is the list size), if one list is the reverse of the other. Often Kendall tau distance is normalized to [0,1] since it can be interpreted as a correlation measure. Therefore, $D(y, z) = 0$ if and only if i and j are in the same order, $D(y, z) = 1$ if and only if i and j are in the opposite order.

Now the objective of SSL label ranking problem is try to predict the rankings for $\hat{y}_{l+1}, \hat{y}_{l+2}, \dots, \hat{y}_{l+u+p}$. A basic assumption is that the similarities lie in the features would also indicate some similarities in different rankings. Based on this assumption, we will detail three models in the following parts.

2.2 Information Gain Decision Tree

Decision tree or a tree-like graph is one of the most popular and practical method in machine learning and data mining. Two motivations of our paper to utilize decision tree are the convenience of getting result and easy for chasing error. The induction of a decision tree is an iteration process. By calculating some splitting criteria, the whole data set can be partitioned into two sub groups. The partition will not suspend until some stopping standard is satisfied. Therefore, we will discuss three sub-problems of the algorithm in the following three aspects.

(i) Find the candidates as splitting node for continuous attributes. C4.5 [10] creates a threshold and then splits the list into those whose attribute value is above the threshold and those are less than or equal to it. To select this threshold, first we sort the data for each dimension, and then determine adjacent examples whose rankings are very dissimilar with each other. Then a series of candidate thresholds can be selected with the median value of the adjacent examples.

(ii) Information gain is one of the most popular method to evaluate a splitting rule which requires to assemble the number of instances belongs to each class. It is straightforward for the labeled data. However,

incomplete data (partial and unlabeled) also involve some information should not be discarded. To utilize this part of information, we want to find the probability of it belongs to every possible permutation. Here we use $\{num_i\}_{i=1}^n$ and $\{mean_i\}_{i=1}^n$ to denote the number of different rankings and the average value of instances for each class. We use $\{z(i)\}_{i=1}^n$ to denote complete rankings in order to judge whether a partial ranking is conflict with complete rankings. Here conflict means there are no discordant pairs between two rankings. (iii) Another

Algorithm 1 Probability of Incomplete Data Belongs to Possible Permutations

Require:
Unlabeled data $U = \{x_i\}_{i=l+1}^{l+u}$. Partial data $P = \{x_i, y_i\}_{i=1+u+1}^{l+u+p}$. The number of different classes n, threshold θ . $\{z(i)\}_{i=1}^n$. $\{mean_i\}_{i=1}^n$. $\{num_i\}_{i=1}^n$.

1. **for** each incomplete data in $x_i, y_i \in U, P$ **do**
   if $num(y_i) \neq 0$ **then**
      if $sim(x_i, mean(y_i)) > \theta$ **then**
         $num(y_i) \leftarrow num(y_i) + 1$. $mean(y_i) \leftarrow (mean(y_i) + x_i)/2$
      end if
   else
      if $z(i)$ and y_i are not conflict **then**
         $num(y_i) \leftarrow num(y_i) + 1$. $mean(y_i) \leftarrow (mean(y_i) + x_i)/2$
      end if
   end if
2. **end for**
3. return updated $\{num\}$ and $\{mean\}$

challenge in our algorithm is the stopping criteria. A generalized common sense of splitting the source is to make the subsets as dense as possible. Therefore, if all instances are not conflict with each other, then the tree is viewed pure enough to stop training. Besides, to prevent over fitting, the building process should also be stopped when the number of instances is small enough.

2.3 Instance Based Learning

The Mallows model [3] has been utilized to solve the label ranking problem, which belongs to the exponential family. Given the model parameters z and θ, the probability of y can be expressed as follows:

$$P(y|\theta, z) = \frac{1}{\Phi(\theta, z)} \exp\left(-\theta D(y, z)\right), \tag{2}$$

where $\Phi(\theta, z) = \sum_{y \in Y} \exp\left(-\theta D(y, z)\right)$ is a normalization constant. z is the distribution's model or center ranking, and $\theta \geq 0$ refers to the dispersion degree.

In this sense, for each incomplete ranking y, we use $E(y)$ to denote all possible permutations which are not conflict with y. By further assuming that the independence of the observations, the probability of y given neighbors $\{x_i, y_i\}_{i=l}^k$ with the parameter z and θ becomes:

$$P(y|\theta, z) = \prod_{i=1}^{k} P\left(E(y_i)|\theta, z\right) = \frac{\prod_{i=1}^{k} \sum_{y \in E(y_i)} \exp(-\theta D(y, z))}{\left(\prod_{j=1}^{n} \frac{1-\exp(-j\theta)}{1-\exp(-\theta)}\right)^k}. \tag{3}$$

Due to the difficulty in deriving the parameter z and θ, a modified Expectation Maximum [5] algorithm can be utilized to solve this problem. Starting from an initial center ranking $z \in Y$, the label information is estimated for the incomplete data by comparing the distance between each possible extension and the center z (E-step). And in the M-step, compute the new center $\hat{z}$ of the distribution. The two steps should be repeated until the center will not change at all. The final center will output as the prediction ranking for the query x.

2.4 Mallows Model Decision Tree

Based on the instance based model, another decision tree algorithm can be utilized to solve this problem. Mallows model has two parameters, the center z and the dispersion parameter θ. By assuming both two sub branches T_{left} and T_{right} follow the Mallows model, we can get θ_{left} and θ_{right} as the estimation of the dispersion

degree. Besides, the size of tree is another crucial criterion. Therefore, we should balance against the purity and the size, a tradeoff standard is to maximize

$$\frac{|T_{left}| \cdot |\theta_{left}| + |T_{right}| \cdot |\theta_{right}|}{|T|}, \tag{4}$$

where $\theta_{left}, \theta_{right}$ denotes the estimated parameters and $|T_{left}|, |T_{right}|, |T|$ denotes the size of corresponding tree.

3 Experiments

In the experiments part, we apply those three algorithms under the SSL framework. A transductive inference method for SSL has been utilized, thereby our job is to predict the label information for the unlabeled data and partial data. The task of experiments is to observe the accuracy and efficiency of those three classifiers: Instances Based Learning (IBL), Mallows Model Decision Tree (MMDT) and Information Gain Decision Tree (IGDT). In this sense, we first partition each experiment into several parts in terms of the number of labeled data. Then in each part, by continuous increasing partial data , we compare the performance variation comparing with supervised learning result.

Considering lacking labeled ranking data, we use the multi-class and regression data sets downloaded from UCI repository to imitate the benchmark data. For classification data, a Naive Bayes classifier is first trained on the complete data. All the labels are then ordered with respect to the predicted class probabilities for each instance. For regression data, a certain number of features is removed from the set of attributes. All attributes are standardized and ordered by size to obtain the ranking. The value of every attribute is scaled into $[-1, 1]$ and the cosine similarity between any two profile vectors is calculated. Some characteristics have been listed in Table 1.

Table 1. Data sets and their properties

data set	attribute	♯inst.	♯attr.	♯labels
iris	real	150	4	3
glass	real	214	9	7
vehicle	real	1518	18	4
concrete	real	1030	9	3
abalone	real	4177	8	3

Some parameters should be set before training the algorithm. In order to approximate the real data set as closely as possible, we train the parameters firstly by a ten-fold cross validation on a sub group of labeled data which are selected randomly from the data set and then apply them in the whole data set by assuming that parameters which have the better performance will also work well upon the whole data set. Besides, to simulate the partial and unlabeled data, some labels are removed from the whole ranking randomly in terms of the requirements of different experiments. When the probability of partial data is 0, the data set is involved with labeled data and unlabeled data only. At that scenario, all data are trained under the supervised learning framework. As for the SSL algorithms, we partitioned the remaining data into partial and unlabeled with equal proportion. Other partitioned situations are not shown here due to the limited space.

The summary of results are shown in Table 2. We can see that all three methods under the SSL framework are performed better than supervised learning, which means partial data and unlabeled data can improve the classifier's performance when enough labeled data are not obtainable. Besides, we can see that with the very limited labeled data the two decision tree algorithms are not as good as the instance based learning (IBL) algorithm especially when the number of partial data is not plentiful. We suspect the reasons lie in two aspects: first, for the instance based learning, it will prefer to select the labeled instances as the neighbors to obtain the useful information; second, no matter for IGDT and MMDT, both of them are intended to select an attribute/value pair as the splitting criteria, thereby the information is so limited that it is difficult to select such a proper criteria. However, with more labeled data are added into the data set, IGDT performs better than MMDT and IBL obviously, which means that IGDT can make use of the limited information as much as possible than other two algorithms.

Besides the accuracy, we also record the average time of iris data set under the matlab platform and the Intel(R) Core(TM)2 Duo CPU E7400 @ 2.80 GHz, 3.21 GB of RAM. For the IBL, the average times for supervised learning and SSL are 1.8478(s) and 0.8969(s), and for MMDT, the individual times are 43.0021(s),

Table 2. Performance of the label ranking algorithms

data set	# of labeled	Supervised IBL	SSL IBL	Supervised MMDT	SSL MMDT	Supervised IGDT	SSL IGDT
iris	10	0.5903	0.7126	0.5294	0.6604	0.5741	**0.7382**
	20	0.6861	0.7725	0.6154	0.7109	0.6519	**0.8098**
	30	0.7121	0.8130	0.6519	0.7312	0.7019	**0.8213**
glass	10	0.6366	0.7759	0.6279	0.7514	0.6468	**0.7805**
	20	0.6904	0.8071	0.6592	0.7684	0.6884	**0.8085**
	30	0.7203	0.8230	0.6819	0.7901	0.7012	**0.8139**
concrete	10	0.3922	**0.4380**	0.3760	0.4186	0.3929	0.4231
	20	0.4029	0.4390	0.3869	0.4266	0.4044	**0.4431**
	30	0.4108	0.4476	0.3968	0.4358	0.4264	**0.4875**
vehicle	10	0.5526	**0.6203**	0.5193	0.5870	0.5048	0.5663
	20	0.5625	**0.6591**	0.5209	0.6072	0.5595	0.5921
	30	0.5772	**0.7010**	0.5296	0.6179	0.5896	0.6394
abalone	10	0.4821	0.7223	0.4606	0.6925	0.5239	**0.7892**
	50	0.5361	0.7580	0.4948	0.7342	0.5596	**0.8193**
	100	0.5372	0.7739	0.5093	0.7491	0.5887	**0.8287**

21.8474(s) and for IBDT, the average times are 2.9855(s), 2.6020(s). Both decision tree algorithms cost longer time than instance based learning. Instance based learning is a lazy classifier, which need little time in the building process while much more time in the prediction procedure. For the two decision tree algorithms, the process of tree induction is a little time consuming especially when confronting more features and labels data set. Besides, our proposed IGDT runs faster than MMDT conspicuously, the reason of which is that MMDT depends on continuous iterations while IGDT depends on simple calculations.

In general, SSL is superior to supervised learning for label ranking. Besides, considering those three algorithms, IBDT performs better than IGDT and IBL in most cases especially when more labeled data are available.

4 Conclusions and Future Work

In this paper, we study the problem of label ranking, which is an efficient approach for object recognition, document classification, recommendation task. Particularly, we introduce semi-supervised learning framework for the first time. We also propose a novel IGDT algorithm. There are still various aspects in further studying the algorithm. 1) Extending the current SSL framework to more algorithms. 2) Trying more distance measurement of rankings, Kendall tau coefficient does not consider diverse significance of different labels in a ranking.

5 Acknowledgment

The work described in this paper is supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No.: CUHK 4128/08*E*).

References

1. A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, page 100. ACM, 1998.
2. M. Boutell, J. Luo, X. Shen, and C. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.
3. W. Cheng, J. *Hühn*, and E. Hüllermeier. Decision tree and instance-based learning for label ranking. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 161–168. ACM, 2009.
4. O. Dekel, C. Manning, and Y. Singer. Log-linear models for label ranking. *Advances in Neural Information Processing Systems*, 16, 2003.
5. A. Dempster, N. Laird, D. Rubin, et al. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

6. P. Diaconis and R. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2):262–268, 1977.
7. E. *Hüllermeier*, J. *Fürnkranz*, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16-17):1897–1916, 2008.
8. K. Huang, Z. Xu, I. King, and M. Lyu. Semi-supervised learning from general unlabeled data. In *Eighth IEEE International Conference on Data Mining, 2008. ICDM'08*, pages 273–282, 2008.
9. M. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81, 1938.
10. J. Quinlan. Improved use of continuous attributes in C 4.5. *Journal of Artificial Intelligence Research*, 4(1):77–90, 1996.
11. C. Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 100(3):441–471, 1987.
12. S. Vembu and T. Gartner. Label ranking algorithms: A survey. *Preference Learning. Springer*, 2009.
13. X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Machine Learning International Workshop then Conference*, volume 20, page 912, 2003.