# A Taxonomy of Label Ranking Algorithms

Yangming Zhou[a,b], Yangguang Liu[a,*], Jiangang Yang[a], Xiaoqi He[a], Liangliang Liu[a]
[a] Ningbo Institute of Technology, Zhejiang University, Ningbo, 315100, Zhejiang Province, China
[b] Department of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, Zhejiang Province, China
*Corresponding author. Email address: ygliu@acm.org

*Abstract*— **The problem of learning label rankings is receiving increasing attention from machine learning and data mining community. Its goal is to learn a mapping from instances to rankings over a finite number of labels. In this paper, we devote to giving an overview of the state-of-the-art in the area of label ranking, and providing a basic taxonomy of the label ranking algorithms. Specifically, we classify these label ranking algorithms into four categories, namely decomposition methods, probabilistic methods, similarity-based methods, and other methods. We pay particular attention to the latest advances in each. Also, we discuss their strengths and weaknesses, and highlight some interesting challenges that remain to be solved.**

*Index Terms*— **label ranking, classification, multilabel learning, rank correlation**

## I. Introduction

Among the problems in the realm of preference learning, label ranking has probably received the most attention in the machine learning literatures in recent years [1]–[3]. Label ranking studies a mapping from instances to rankings over a finite number of predefined labels. It can be considered as a variant of the conventional classification problem, where only a single label is requested instead of a ranking of all labels.

There are a large number of practical applications of label ranking [3]–[5], in which the target is to learn an exact label preference of an instance in form of a complete ranking. For example, in bioinformatics, where the task is to rank a set of genes according to their expression level (measured by microarray analysis) based on features of their phylogenetic profile [4]. Another interesting application scenario is metalearning, where the target is to induce a total rank of available algorithms according to their suitability for a new dataset, based on the characteristics of the dataset [3], [6].

A number of label ranking methods have been proposed for label ranking learning, some of which consist of adapting existing classification algorithms (e.g., k-nearest neighbor [6], decision trees [7]). Some approaches are based on probabilistic models, such as Mallows model [8] and Plackett-Luce (PL) model [9]. Some approaches take advantage of the similarities of the rankings, such as naive Bayes for label ranking [3]. Other methods including rule-based label ranking [10].

A great deal of effort has been made on label ranking. What seems to be lacking, however, is an overview of the existing approaches to label ranking. One exception can be found in [11], where the authors investigate a plethora of label ranking algorithms. However, they focus on several supervised learning problems including multiclass classification, multilabel classification and hierarchical classification. As same as their work, we hope that by giving an overview of existing literatures on label ranking, the reader would be able to capture the current research status and directions of development in this field. In addition, a considerable number of new methods for label ranking have been proposed in the past two or three years, such as multilayer perceptron [12], association rule [13], and Gaussian mixture model (GMM) [14] for label ranking. Consequently, it is necessary to re-conduct a systematic literature review of label ranking algorithms.

The main aim of this review is to make interested reader aware of the possibilities of label ranking, and provide a taxonomy of label ranking techniques, focusing on the existing literatures. We conduct a comprehensive overview of the state-of-the-art label ranking algorithms. In summary, the major contributions of this paper are as follows:

1) we review the currently existing major label ranking algorithms, providing a basic taxonomy of these algorithms.
2) we summarize the main performance metrics used for evaluating label ranking algorithms.
3) we discuss their benefits and drawbacks, and highlighting some interesting challenges that remain to be solved.

The rest of the paper is organized as follows. In section II, we recall the problem of the label ranking in a more formal setting, discuss the difference between ranking and classification problem, and summarize the performance metrics used in the label ranking. Then, in section III, we introduce and analysis the major label ranking algorithms and we present a taxonomy of these label ranking approaches. Finally, conclusions and some interesting challenges that remain to be solved are presented in section IV.

## II. Preliminaries

### A. Problem Description

A general setting of the label ranking problem given here follows the one provided by Cheng *et al* [9]. Label ranking can be seen as an extension of the conventional setting of classification. Roughly speaking, the former is


doi:10.4304/jcp.9.3.557-565

obtained from the latter through replacing single class label by a total order of all class labels. Instead of predicting every given instance $x$ from an instance space $\mathbb{X}$ with one class label $\lambda$ among a finite set of class labels $\mathcal{L} = \{\lambda_1, \ldots, \lambda_n\}$, we associate $x$ with a order of all class labels, i.e., a complete, transitive, and asymmetric relation $\succ_x$ on $\mathcal{L}$, where $\lambda_i \succ_x \lambda_j$ indicates that $\lambda_i$ precedes $\lambda_j$ in the rank associated with $x$.

Formally, a ranking $\succ_x$ can be identified with a permutation $\pi_x$ of $\{1, 2, \ldots, n\}$. It is convenient to define $\pi_x$ such that $\pi_x(i) = \pi_x(\lambda_i)$ is the position of the label $\lambda_i$ in the ranking. A complete ranking for the set $\mathcal{L}$ is therefore denoted as:

$$\lambda_{\pi_x^{-1}(1)} \succ_x \lambda_{\pi_x^{-1}(2)} \succ_x \cdots \lambda_{\pi_x^{-1}(n)} \tag{1}$$

where $\pi_x^{-1}(j)$ is the index of the label at position $j$ in the ranking. The class of permutations of $\{1, 2, \ldots, n\}$ is denoted by $\Omega$. By abuse of terminology, we shall refer to element $\pi \in \Omega$ as both permutations and rankings.

Specifically, in analogy with the classification setting, each instance is associated with a probability distribution over $\Omega$. That is, for every instance $x \in \mathbb{X}$, there exists a probability distribution $P(\cdot|x)$ such that, for every $\pi \in \Omega$, $P(\pi|x)$ is the probability that $\pi_x = \pi$.

The goal in label ranking is to learn a "label ranker" in the form of an $\mathbb{X} \to \Omega$ mapping. As training data, a label ranker has assess to a set of training instances $x_i$, $i = 1, 2, \ldots, m$, together with information about the associated ranking $\pi_{x_i}$. In practice, we do not have access to the complete ranking of all class labels, but only partial information about their rankings. In order to conform with the practical applications, however, it is important to allow for partial preference information. The incomplete label ranking is denoted as:

$$\lambda_{\pi_x^{-1}(i_1)} \succ_x \lambda_{\pi_x^{-1}(i_2)} \succ_x \ldots \succ_x \lambda_{\pi_x^{-1}(i_{\hat{n}})} \tag{2}$$

where $\{i_1, i_2, \ldots, i_k\}$ is a subset of the index set $\{1, 2, \ldots, n\}$ such that $1 \leqslant i_1 < i_2 < \ldots < i_{\hat{n}} \leqslant n$. For example, given an instance $x$, it might be known that $\lambda_3 \succ_x \lambda_1 \succ_x \lambda_4$, while no preference information is given about $\lambda_2$ and $\lambda_5$.

### B. Ranking and classification

Label ranking is a very interesting problem as it associates with three important supervised learning problems, including multiclass classification, multilabel classification and multilabel ranking. In the following, we illustrate the differences among these four learning problems in Figure 1.

- Multiclass classification is considered as one of the most important supervised learning task, where each instance is associated with a single label from a set of disjoint labels (see Figure 1(a)). As we mentioned above, existing methods for label ranking are typically extension of conventional classification algorithms.
- Multilabel classification concentrates on learning a model that output a bipartite partition of all class labels into relevant and irrelevant labels with respect to a query instance (see Figure 1(b)). The significant difference with respect to multiclass classification is that each instance is associated with a subset of labels, instead of a single label.
- Label ranking on the other hand extends multiclass classification in the sense that it is devote to giving an ordering of all class labels (see Figure 1(c)). Besides, It has been observed by researchers [1], [2], [15] that many conventional supervised learning problems, such as multiclass classification and multilabel classification, can be formalized in terms of label ranking.
- Multilabel ranking is a complex prediction problem where the goal is to not only identify relevant labels from a set of predefined class labels, but also to rank them according to their relevance to a query instance (see Figure 1(d)). Consequently, multilabel ranking can be considered as a generalization of multilabel classification and label ranking.

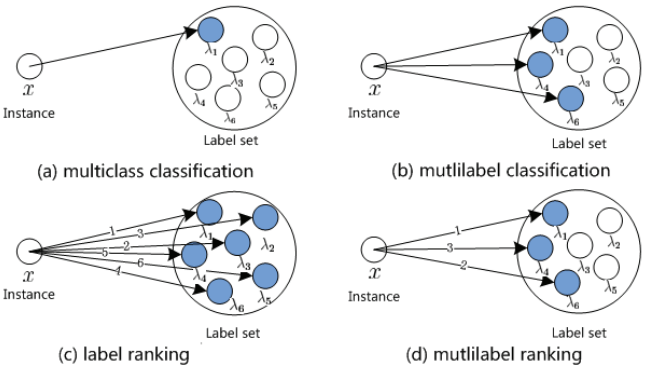


Figure 1. Schematic illustration of four supervised learning problems: (a) multiclass classification where each instance is labeled with a single class label $\{\lambda_1\}$. (b) multilabel classification where each instance is annotated with multiple class labels $\{\lambda_1, \lambda_4, \lambda_6\}$. (c) label ranking where each instance is associated with a total order of all class labels $\{\lambda_1 \succ \lambda_3 \succ \lambda_2 \succ \lambda_6 \succ \lambda_4 \succ \lambda_5\}$. (d) multilabel ranking where each instance is linked to an ranking of the possible class labels $\{\lambda_1 \succ \lambda_6 \succ \lambda_4\}$.

Unlike conventional classification, label ranking sorts the class label $\lambda_i$ according to their conditional class probabilities $p(\lambda_i|x)$ instead of assuming the existence of a "true class label" of an instance. In fact, the former can be seen as a special case of the latter, the other way around, the latter actually can be interpreted as a generalization of the former. Roughly speaking, it is obtained by associating the "true class" in classification with the top-ranked label in label ranking [16].

### C. Evaluation measures

To evaluate the accuracy of the predicted ranking $\pi$ relative to the corresponding target ranking $\pi_0$, a suitable accuracy measure defined on the permutation space $\Omega$ over $n$ is needed. There are two main accuracy measures for evaluating the performance of label ranking algorithms, i.e., Spearman's rank and Kendall's tau correlation coefficients.

*Spearman's rank* is a popular and widely accepted similarity metric for rankings [17]. It is originally proposed as a non-parametric rank statistic to measure the strength of the association between two variables. Formally, it is defined as:

$$\rho = 1 - \frac{6D(\pi_0, \pi)}{n(n^2 - 1)} \tag{3}$$

where $D(\pi_0, \pi)$ denotes the sum of squared rank distances, i.e., $\sum_i (\pi_0(i) - \pi(i))^2$.

*Kendall's tau* is another alternative similarity metric for rankings [18]. This measure essentially is a linear transformation of the number of discordant pairs of labels of two rankings. It is defined as:

$$\tau = 1 - \frac{4D(\pi_0, \pi)}{n(n - 1)} \tag{4}$$

where $D(\pi_0, \pi)$ denotes the number of discordant pairs of labels in target ranking $(\pi_0)$ and predicted ranking $(\pi)$, i.e., $\sum_{(i,j):\pi_0(i)>\pi_0(j)} [\pi(i) < \pi(j)]$.

Due to their inherent simplicity, similarity measures still remain prominent in evaluating the new label ranking algorithm. However, these measures fail to take into account some facts in information retrieval or recommender systems. For example, an error on a highly-relevant label should result in a high penalty than an error on a low-relevant label. In addition, errors at the top of the rank should be costlier than errors at the tail of the rank. Some extensions of conventional distance measures have been proposed, which take into account both label relevance and positional information [19].

## III. Label ranking algorithms

Various label ranking algorithms have already been proposed in recent two or three years [3], [10], [12], [13]. We devote to reviewing the label ranking algorithms proposed in recent years, and divided them into four categories (see the Table I), i.e., decomposition methods, probabilistic methods, similarity-based methods, and other methods. For each algorithm, the core idea of the algorithm, as well as its strengths and weakness will be discussed in the following. A detail description of these label ranking algorithms is beyond the scope of this paper.

TABLE I.
SUMMARIZES OF EXISTING LABEL RANKING ALGORITHMS
(INCOMPLETE LIST)

| Category | Label ranking methods | References |
|---|---|---|
| Decomposition | Constraint classification | [20] |
| | Log-linear model | [1] |
| | Ranking by pairwise comparison | [2], [15], [21] |
| Probabilistic | Instance-based (Mallows) | [8], [22], [23] |
| | Decision trees | [7], [22] |
| | Instance-based (Plackett-Luce) | [9] |
| | Generalized linear models | [9] |
| | Gaussian mixture model | [14] |
| Similarity | Naive Bayes | [3] |
| | Association rules | [13] |
| | Multilayer perceptron | [12] |
| Others | Rule-based | [10] |

### A. Decomposition methods

Decomposition technique is an efficient way to solve complex prediction problems, where a label ranking problem is decomposed into several simpler sub-problems, usually binary classification problems, and then the solutions of these sub-problems are combined into output rankings. Subsequently, we outline three main label ranking algorithms in the category of decomposition methods, including constraint classification (CC), log-linear models for label ranking (LL), and ranking by pairwise comparison (RPC).

*Constraint classification:* It turns the original problem into a single binary classification problem in an expanded, high-dimensional space, and constructs a label ranking model from the classifier learned in that space [20]. More specifically, learning the functions $f_i(\cdot), i = 1, 2, \ldots, n$, from training data. The linear utility functions are expressed as:

$$f_i(x) = \sum_{k=1}^{d} \omega_{ik} x_k \tag{5}$$

where $\omega_{ik}, k = 1, 2, \ldots, d$, denotes label-specific coefficients, a preference relation $\lambda_i \succ_x \lambda_j$ translates into the constraint $f_i(x) - f_j(x) > 0$ or, equivalently, $f_j(x) - f_i(x) < 0$. Both positive constraint and negative constraint can be expressed in terms of the sign of an inner product $\langle z, \omega \rangle$, where $\omega = (\omega_1, \omega_2, \ldots, \omega_n)$ and $\omega_i = (\omega_{(i-1)d+1}, \ldots, \omega_{id})$ is the $i$-th chunk of the weight vector $\omega \in \mathbb{R}^{nd}$. Correspondingly, the vector $z$ is constructed by mapping the original $d$-dimensional training example $x$ into an $n \times d$-dimensional space. More specifically, for the positive constraint, we constructed an new $n \times d$-dimensional vector $z = (0^T, 0^T, \ldots, x^T, \ldots, -x^T, \ldots, 0^T)$. That is, they are block vectors having zeros everywhere except at the $i$-th and $j$-th block position, where they have $x$ and -$x$, respectively, for $i \neq j$. For the negative constraint, a vector is generated with the same elements but reversed signs. Both constraints can serve as training examples for a conventional binary classifier in an $n \times d$-dimensional space. The task now is to find a separating hyperplane in this space, that is, a suitable vector $\omega$ satisfying all constraints. Consequently, for an new instance $x$, the ranking of all labels is given by $\arg sort_{j \in \{1,\ldots,n\}} \langle \omega_j, x \rangle$. As this method works solely in an inner product space, it can be kernelized when more complex utility functions are desired [24].

*Log-linear model for label ranking:* The key idea of this approach is also to learn a utility function for each individual label [1]. Here, utility functions are expressed as in terms of linear combination of a set of base ranking functions:

$$f_i(x) = \sum_j \omega_j h_j(x, \lambda_i) \tag{6}$$

where $h_j(\cdot)$ is a base ranking function, which maps instance/label pairs to real numbers. A boosting based algorithm is proposed to estimate the model parameters $\omega_j$, which seeks to minimize a ranking error in an iterative

way [1]. In a special case, when the base functions are defined as follow

$$h_{kj}(\mathbf{x},\lambda)=\begin{cases}x_k & \lambda=\lambda_j\\ 0 & \lambda\neq\lambda_j\end{cases}(1\leqslant k\leqslant d, 1\leqslant j\leqslant n) \quad (7)$$

the approach is essentially equivalent to constraint classification [2], and it equal to learning class-specific utility function (5). In this method, there is not strict assumption about a set of preferences over the label-set. This enables the method to accommodate a variety of ranking problems.

*Ranking by pairwise comparison:* This method attempts to model the individual preferences directly instead of translating them into a utility function. It is an extension of the well-known reduction technique known as pairwise classification [25]. The key idea of ranking by pairwise comparison is to transform a $n$-label ranking problem into $n(n-1)/2$ binary problems, one for each pair of labels. More specifically, a separate model $\mathcal{M}_{i,j}$ is trained for each pair of labels $(\lambda_i,\lambda_j)\in\mathcal{L}\times\mathcal{L}, 1\leqslant i<j\leqslant n$, using the training examples for which either $\lambda_i\succ_x\lambda_j$ or $\lambda_j\succ_x\lambda_i$ is known. Then, given a instance $x\in\mathbb{X}$, $\mathcal{M}_{i,j}$ is intended to learn a mapping that outputs 1 if $\lambda_i\succ_x\lambda_j$, and 0 if $\lambda_j\succ_x\lambda_i$. This mapping can be realized by any binary classifier. Alternative, one may also learn a model that maps into the unit interval $[0,1]$ instead of $\{0,1\}$. Typically, a valued preference raltion $\mathcal{R}_x$ is assigned to every query instance $x\in\mathbb{X}$.

$$\mathcal{R}_x(\lambda_i,\lambda_j)=\begin{cases}\mathcal{M}_{i,j}(x) & \text{if } i<j,\\ 1-\mathcal{M}_{i,j}(x) & \text{if } i>j\end{cases} \quad (8)$$

for all $\lambda_i\neq\lambda_j\in\mathcal{L}$. Usually, the output of binary classifier can be interpreted as a probability, i.e., the closer the output of $\mathcal{M}_{ij}$ to 1, the stronger the preference $\lambda_i\succ_x\lambda_j$ is supported [21].

Finally, an associated ranking $\pi_x$ based on the results of these individual models $\mathcal{M}_{i,j}$ can be obtained by using a ranking procedure. A simple though effective strategy is a generalization of voting strategy: each label $\lambda_i$ is assigned a score by the sum of votes $S(\lambda_i)=\sum_{\lambda_j\neq\lambda_i}\mathcal{R}_x(\lambda_i,\lambda_j)$, and all labels are then ordered according to these scores, i.e., $(\lambda_i\succ_x\lambda_j)\Rightarrow(S(\lambda_i)\geqslant S(\lambda_j))$. This strategy exhibits desirable properties such as transitivity of pairwise preferences. Furthermore, the RPC algorithm has some competitive advantages. For example, the modular conception of RPC allows for combining different learning and ranking methods in a convenient way (i.e., different loss functions can be minimized by simply changing the ranking procedure but without the need to retrain the binary models ) [2].

*Benefits and drawbacks:* Even though these methods have shown good performance in experimental studies [2], the decomposition of the complex label ranking problem to the simple binary classification problem is not self-evident and does not come for free. Such decomposition becomes possible only through the use of an ensemble of binary models; in CC and LL, the size of this ensemble is linear in the number of labels, while in RPC it is quadratic. In practice, RPC can be expected to be computationally more efficient than alternative approaches like CC. Specifically, the total number of training examples constructed by RPC is no more than $m\cdot(n(n-1)/2)$, while CC constructs twice as many training examples as RPC, i.e., $m\cdot(n(n-1))$. Some problems come along with such an ensemble. For example, theoretical assumptions on the sought "ranking-valued" mapping, which may serve as a proper learning bias, may not be easily translated into corresponding assumptions for classification problems [9]. This is especially true for methods like CC, where the transformation from rankings to classification strongly exploits the linearity of the underlying utility functions. It has been shown that it is often not clear (and mostly even wrong) that minimizing the classification error, or a related loss function on the binary problems is equivalent to maximizing the performance of the label ranking model in terms of the desired loss function on rankings [26].

### B. Probabilistic methods

The methods, which avoid the aforementioned problems to some extent, were recently proposed in [9], [14], [23]. The key idea is to develop label ranking method on the basis of statistical models for ranking data. More specifically, it introduces a probability model to estimate predictive models for ranking. In statistic, there have numerous different types of probability distribution on ranking. The two most popular in the label ranking community are the Mallows model [27] and the Plackett-Luce model [28], [29]. In addition, gaussian mixture model is also introduced to deal with the label ranking problem. In the following, we will respectively outline label ranking methods based on these three probability models.

*Mallows model for label ranking :* The Mallows model is a distance-based probability model, which defines the probability of a ranking according to its distance to a center ranking [27]. Since many permutation distance measures are available, the Mallows model shows strong expressiveness. The standard Mallows model is defined as follows

$$P(\pi|\theta,\pi_0)=\frac{\exp(-\theta D(\pi,\pi_0))}{\phi(\theta)} \quad (9)$$

The ranking $\pi_0\in\Omega$ is the location parameter (center ranking) and $\theta\geqslant 0$ is a spread parameter. Moreover, $D(\cdot)$ is a distance measure on rankings and $\phi(\theta)$ is the normalization constant that dependent on the spread parameter $\theta$. It is quite apparent that the model assigns the maximum probability to the center ranking $\pi_0$. The larger the distance metric $D(\pi,\pi_0)$, the smaller the probability $P(\pi|\theta,\pi_0)$ becomes [23].

Two commonly used classification methods, namely instance-based learning and decision tree induction have been adapted for label ranking learning [22]. Both methods are based on local estimation principles and known to have a rather weak bias. Consider a query instance

$x \in \mathbb{X}$ and let $x_1, x_2, \ldots, x_k$, denote the $k$ nearest neighbors of $x$ (according to the Euclidean distance) in the training set. Each neighbor $x_i, i = 1, 2, \ldots, k$, is associated with a ranking $\pi_i \in \Omega$. Generally, it is assumed that the probability distribution $P(\cdot|x)$ on $\Omega$ is (at least approximately) locally constant around the query $x$. By furthermore assuming that the rankings have been generated independently of each other by the Mallows model. Consequently, the probability to observe $\pi = \{\pi_1, \pi_2, \ldots, \pi_k\}$ given the parameter$(\theta, \pi_0)$ becomes

$$\mathbf{P}(\pi|\theta, \pi_0) = \prod_{i=1}^{k} \mathbf{P}(\pi_i|\theta, \pi_0) \tag{10}$$

The maximum likelihood estimation (MLE) of parameters $(\theta, \pi_0)$ are then given by those parameters that maximize the probability (10).

Consider the more general case of incomplete ranking information, which means that a ranking $\pi_i$ does not necessarily contain all labels. Computing the MLE of $(\theta, \pi_0)$ now becomes more difficult. Cheng *et al* [23] proposed an approximation procedure, which replaces the E-step of EM (expectation-maximization) algorithm by another maximization step. Specifically, given $\hat{\pi}_0$ as an initial center ranking, each incomplete neighbor ranking $\pi_i$ is replaced by the most probable consistent extension (first M-step). Having replaced all neighboring rankings by their most probable consistent extensions[1], an MLE $(\hat{\theta}, \hat{\pi})$ can be derived as described for the case of complete information above (second M-step).The center ranking $\hat{\pi}_0$ is then replaced by $\pi_0$, and the whole procedure is iterated until the center does not change any more; $\pi^*$ is then output as a prediction [23].

Likewise, decision tree has also been used for label ranking learning. The main modifications of conventional decision tree learning concerns the split criterion at inner nodes and the criterion for stopping the recursive partitioning. More detail can be found in [22] and will not fully detailed here.

Both methods are based on local estimation principle and are known to have a rather weak bias. These are especially useful for problems requiring complex decision boundaries. It has been proved by empirical studies, instance-based label ranking is particularly competitive in terms of predictive accuracy, while decision trees are often praised for their good interpretability [9], [22]. Besides, both methods provide natural measure of the reliability of a prediction. Notwithstanding some appealing properties, it is not ideal to handle incomplete training data. Roughly speaking, incomplete observations will greatly increase the computational complexity of maximum likelihood estimation about $\pi_0$ and $\theta$. In general, it requires a time complexity of $\mathcal{O}(n!)$ to compute the probability of a single permutation of n labels [30].

*Plackett-Luce model for label ranking:* Plackett-Luce (PL) model [28], [29] is an alternative to Mallows model. It is highly competitive to start-of-the-art methods in terms of predictive accuracy, especially in learning from possibly incomplete ranking information. The Plackett-Luce model is specified by a parameter vector $\mathbf{v} = (v_1, v_2, \ldots, v_n) \in \mathbb{R}_+^n$:

$$\mathbf{P}(\pi|\mathbf{v}) = \prod_{i=1}^{n} \frac{v_{\pi(i)}}{\sum_{j=i}^{n} v_{\pi(j)}} \tag{11}$$

The PL model is a stagewise model, which decomposes the process of generating a ranking of $n$ labels into $n$ sequential stages. At the $i$-th stage, selecting one from the labels that have not been selected so far, then assigned to the position $i$ according to a probability based on the scores of the unassigned labels. The product of the selection probabilities at all the stages defines the probability of the ranking [30]. It is easy to verify that the probability of an incomplete ranking (2) is given by

$$\mathbf{P}(\pi|v) = \prod_{i=1}^{\hat{n}} \frac{v_{\pi(i)}}{\sum_{j=i}^{\hat{n}} v_{\pi(j)}} \tag{12}$$

It is clearly that the probability of complete ranking and incomplete ranking have exactly the same form, except that the number of factors is $\hat{n}$ (the number of labels observed) instead of $n$.

Two methods based on the PL model including instance-based and generalized linear method for label ranking have been proposed in [9]. The key idea of first method is to use the PL model to fit locally constant probability model in the context of instance-based learning. Given the parameters $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, the probability to observe the rankings $\pi = \{\pi_1, \pi_2, \ldots, \pi_k\}$ in the neighborhood becomes

$$P(\pi|\mathbf{v}) = \prod_{i=1}^{k} \prod_{m=1}^{\hat{n}_i} \frac{v_{\pi_i(m)}}{\sum_{j=m}^{\hat{n}_i} v_{\pi_i(j)}} \tag{13}$$

where $\hat{n}_i \in \{2, \ldots, n\}$ represents the number of labels ranked by $\pi_i$. The MLE of $\mathbf{v}$ is then given by those parameters that maximize the probability (13) or, equivalently, the log-likelihood function. Minorization Maximization [31] is a powerful iterative algorithm to find the MLE parameters of the PL model. This procedure provably converges to an MLE estimation of the PL parameters.

Given the MLE parameter vector $\mathbf{v}^*$, a prediction of the ranking associated with $x$ can be derived from the distribution $\mathbf{P}(\cdot|\mathbf{v}^*)$ on $\Omega$. That is, a ranking with the highest posterior probability

$$\pi^* \in \arg\max_{\pi \in \Omega} \mathbf{P}(\pi|\mathbf{v}^*) \tag{14}$$

A desired ranking can be easily obtained by ordering the labels $\lambda_i$ in decreasing order according to corresponding parameters $v_i^*$, i.e., for all $1 \leq i < j \leq n$, we have

$$v_{\pi^*(i)} \geq v_{\pi^*(j)} \tag{15}$$

In fact, due to the special structure of the probability distribution (11), a ranking of the form (15) is not only the most intuitive prediction, but also provably optimal for almost all common loss functions on rankings [9].

[1]A permutation $\pi \in \Omega$ is a consistent extension of $\pi_i$ if it ranks all labels including in ranking $\pi_i$ in the same order

In contrast with instance-based learning based on PL model, the second approach estimates a global model instead of a local model. The central idea of this method is to define the PL parameters $\mathbf{v}$ as a function of the instance [9]. More precisely, the logarithm of each parameter $v_i$ is modeled as a linear function $v_i = \exp(\langle \omega_i, x_j \rangle) = \exp(\sum_{q=1}^{d} \omega_{iq} \cdot x_{jq})$, where an instance is represented in terms of a feature vector $x_j = (x_{j1}, x_{j2}, \ldots, x_{jd}) \in \mathbb{X} \subseteq \mathbb{R}^d$. Consequently, the model parameters to be estimated are $\omega_{ij} (i = 1, 2, \ldots, n, j = 1, 2, \ldots, d)$. Given a training dataset $\mathcal{T} = \{(x_i, \pi_i)\}_{i=1}^{m}$ with $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, then the log-likelihood function is given by

$$L = \sum_{i=1}^{m} \sum_{j=1}^{\hat{n}_i} [\log(v(\pi_i(j), i)) - \log \sum_{l=j}^{\hat{n}_i} v(\pi_i(l), i)] \quad (16)$$

where $\hat{n}_i$ is the number of labels in the ranking $\pi_i$, and $v(j, i) = \exp(\langle \omega_j, x_i \rangle) = \exp(\sum_{q=1}^{d} \omega_{jq} \cdot x_{iq})$. The maximization of the log-likelihood can be accomplished by means of gradient-based optimization methods, such as a standard stochastic gradient descent algorithm [32].

*Gaussian mixture model for label ranking:* This approach could be attractive because of an intuitively clear learning process and ease of implementation. The model consists of mixtures defined by prototypes, which is associated with preference judgment for each pair of labels. Unlike other approaches, gaussian mixture model for label ranking is not limited to a special type of label ranking and could support various ranking structures (bipartite, multipartite, etc).

The GMM model for label ranking solves the preference learning task in two steps. It first learns soft pairwise label preferences via minimization of the proposed soft rank loss measure. Then, these soft label preferences in form of a preference matrix need to be aggregated into a total order of labels. The model is completely defined by a set of $K$ mixtures, i.e., prototype $\{(\mu_k, Q_k), k = 1, 2, \ldots, K\}$, where $\mu_k$ is the prototype position and $Q_k$ is the corresponding preference matrix. Standard supervised learning techniques, such as gradient descent or Expectation Maximization can be used to find the unknown model parameters.

*Benefits and drawbacks:* Instead of simply generating a prediction in terms of a ranking, the probabilistic methods for label ranking allow one to complement predictions by diverse types of statistical information. More specifically, probabilistic methods provide the measure of the reliability of a prediction. Compared with the Mallows model, the PL model has a polynomial time complexity instead of a time complexity $\mathcal{O}(n!)$. Empirical results show that the PL model seems preferable in the case of incomplete ranking information, not only computationally but also performance [9]. However, its drawback is that it is defined on the scores of individual label and can not take advantage of common distance measure between the two rankings [30]. In addition, instance-based label ranking approaches with PL or Mallows model require store the entire training data in memory, which can be costly or even impossible in the source-constrained applications. Grbovic *et al* [14] proposed an online, time- and memory-efficient label ranking approach based on the Gaussian mixture model, which could be attractive because of an intuitively clear learning process and ease of implementation. However, it requires aggregation of the predicted label preference matrix to produce a total order of labels, and the aggregated process has a high complexity.

### C. Similarity-based methods

The key idea of this category method is to replace the concept of probability by the similarity between the rankings. It has been proved that there is a parallel between probabilities and distance measure [33]. That is, maximizing the likelihood is equivalent to minimizing the distance (i.e., maximizing the similarity) in a Euclidean space. In recent years, several articles have been devoted to adapting the existing machine learning methods for label ranking problems based on similarity measures [3], [12], [13]. In the following, we give a brief review of these methods.

*Naive Bayes for label ranking:* Based on the fact that there exists a parallel between the concepts of distance and likelihood, an adaptation of a naive Bayes classification algorithm for label ranking is proposed [3]. Roughly speaking, the difference between classification and label ranking lies in the target variable. Therefore, it is necessary to adapt the parts of the algorithm that depend on the target variable, including prior probability $P(\pi)$ and conditional probability $P(x_i|\pi)$. More specifically, the prior probability of a label ranking is given as $P(\pi) = \sum_{i=1}^{m} \rho(\pi, \pi_i)/m$, where the similarity measure uses the Spearman's rank correlation $\rho$. In fact, the naive Bayes classifier makes one simple, naive, assumption that the attribute values are conditionally independent from each other [34]. This implies that the conditional probability of label ranking can write as $P(x_i|\pi) = \prod_{a=1}^{d} P(x_{i,a}|\pi)$, where $x_{i,a}$ denotes the value $i$ of attribute $a$. Finally, naive Bayes for label ranking obtain the prediction ranking with the higher $P(\pi|x_i)$ value:

$$\hat{\pi} = \arg\max_{\pi \in \Omega} P(\pi|x_i) \quad (17)$$

Experimental results on a number of metalearning datasets show that the adaptation naive Bayes method can be a good solution to metalearning problems [3]. However, its drawback is that it cannot deal with the problem with partial ranking information.

*Association rules for label ranking:* There also exists an adaptation of association rules mining algorithm APRIORI [35] for label ranking based on similarities between the rankings, where the goal is to discover frequent pairs of attributes associated with a ranking [13]. Correspondingly, the label ranking association rules can be defined as $condset \rightarrow \pi$, and $condset \subseteq x$, $\pi \in \Omega$. It means that if an example matches the rule $condset \rightarrow \pi$, then the predicted ranking is $\pi$. In order to make association rule can be applied to label ranking learning, variations of the support and confidence must be made based on

ranking similarities. Given a similarity measure between rankings, i.e., $s(\pi_a, \pi_b)$, the support (sup) of the ruleitem $\langle condset, \pi \rangle$ is defined as follows:

$$sup(\langle condset, \pi \rangle) = \frac{\sum_{i:condset \subseteq x_i} s(\pi_i, \pi)}{m} \quad (18)$$

where $s(\pi_a, \pi_b) = \rho(\pi_a, \pi_b)$ or $\tau(\pi_a, \pi_b)$, if the correlation metrics value (Spearman's rank correlation $\rho$ or Kendall's tau correlation $\tau$) bigger than 0, otherwise, $s(\pi_a, \pi_b) = 0$. The confidence (conf) of a rule $condset \rightarrow \pi$ is obtained as follows:

$$conf(\langle condset, \pi \rangle) = \frac{sup(\langle condset, \pi \rangle)}{sup(condset)} \quad (19)$$

APRIORI algorithm identifies all rules that have a support and confidence higher than a given minimal support threshold and a minimal confidence threshold, respectively [36]. Additionally, a simple greedy method (matching maximization) is used to determine the parameters of the algorithm. Thus, a prediction can be made from a set of label ranking association rules $\mathcal{R}$, which maximize the number of matching examples and contain the best rules. The experimental results clearly show that this is a viable label ranking method [13]. Moreover, it competes well with state-of-the-art label ranking algorithms. Despite the usefulness and simplicity of APRIORI, it runs a time consuming candidate generation process and needs large memory. In addition, it needs multiple scans of the database and typically generates a very large number of rules.

*Multilayer perceptron for label ranking:* In addition to the above-mentioned two label ranking methods, multilayer perceptron (MLP) algorithm also has been adapted to solve label ranking problems based on similarity measures [12]. MLP is a neural network that using back propagation to determine appropriate weights for the connections in the network. To make MLP for label ranking problems, two key issues must be addressed, i.e., the error functions guiding the back-propagation learning process and the method to generate a ranking from the output layer.

Given that there have $d$ nodes in the input layer, each attribute of instance $x$ represents one input signal. Correspondingly, output layer contains $n$ nodes, each one is associated with a label. Output on the $k$-th training example is denoted as $y_j(k)$ for each output node $j$. Finally, ordering the scores assigned to each output layer node $y_j(k), j = 1, 2, \ldots, n$, thus, obtain a predicted ranking $\hat{\pi}_k$. Various approaches have been developed to determine the error signal $c_j(k)$ of $j$-th output nodes [12]. For example, the error signal is defined in terms of the label ranking error $e_\tau(k)$. Namely, $c_j(k) = e_\tau(k) = (1 - \tau(\pi_k, \hat{\pi}_k))/2$, where $\tau(\pi_k, \hat{\pi}_k)$ denotes Kendall's $\tau$ correlation coefficient. Furthermore, the weight correction $\omega_{ji}(k)$ is updated based on the estimated error $c_j(k)$. The correction is defined as follows:

$$\Delta\omega_{ji}(k) = \eta c_j(k) y_i(k) \quad (20)$$

where $\eta$ is the learning rate. Empirical results indicate that MLP for label ranking is likely to complete with other methods, especially when the data contain more attributes [12].

*Benefits and drawbacks:* Similarity-based label ranking approaches have some important properties, they assigned non-zero probabilities to rankings that are not observed in the available data. The greater the similarity between two particular rankings, the higher is the probability that the next observed ranking will be similar to the known ranking. Therefore, it can be good solution to handle the stochastic nature of rankings. Even though have these important advantages there is still scope for future improvements in terms of prediction accuracy. For example, improving the method for prediction generation and matching maximization can achieve better ranking performance of naive Bayes for label ranking method.

### D. Other methods

In addition, there also exist other approaches, which cannot be classified into either of previously defined groups. For example, rule-based label ranking method [10], which is an approach based on a learning reduction technique and provides a predicting model based on decision rules in the form:

$$IF \ \Phi \ THEN \ \Psi \quad (21)$$

where $\Phi$ is called the "antecedent" and $\Psi$ is called the "consequent". $\Phi$ is typically composed of condition on the values of some attributes,usually connected by a logical conjunction operator (AND), while $\Psi$ is generally the rank to which an instance satisfying the antecedent $\Phi$ should be assigned.

*Benefits and drawbacks:* Compared to the aforementioned methods, rule-based method is more appropriate for real-world applications since it does not perform like black box and can give clear and directly interpretable results to practitioners. Additionally, the approach has a modular conception conception, which can be combined with other classifier algorithms (e.g., Decision Trees, SVM), especially in case very large data sets has to be treated. Despite these important advantages there is still scope for future improvements in terms of prediction accuracy and computational time. For example, the size of the training set can drastically increase when the number of labels is large, i.e., the complexity of the algorithm for the inference of rules is polynomial, i.e., $m \cdot (n(n-1)/2)$, where $m$ is the number of training examples in the original data set and $n$ is the number of class labels.

## IV. Conclusions and Future Perspectives

A large and fruitful effort has been performed during the last years in the adaptation and proposal of label ranking methods. In this article, we conducted a review on currently existing label ranking method and provided a basic taxonomy of these methods. We should remark that a detailed description of discussing label ranking methods is beyond the scope of this paper. However, we aim is to provide a basic taxonomy of label ranking techniques,

discuss their benefits and drawbacks, and highlight some interesting challenges that remain to be solved. We hope that the references in this paper cover the major aspect of label ranking problem, and provide a guide to interested readers in this research direction.

The most challenging aspect of label ranking is the possibility of predicting weak or partial orderings of labels. The Mallows model was used in the context of an instance-based approach to label ranking [23]. Despite some appealing properties, it is have been proved that the Mallows model is not ideal for handling incomplete training data [9]. i.e., for $n$ labels ranking problem, the time complexity is $\mathcal{O}(n!)$. In addition, the distance or correlation measures used in the discussing label ranking methods are failed to take into account some important facts [19]. i.e., an error on a highly relevant label should result in a high penalty than an error on a low relevant label. In a similar vein, errors at the top of the rank should be costlier than errors at the tail of the rank.

To deal with these problems, there are several directions for future work. Firstly, it is likely that the prediction of rankings can be improved by combining label relevance and positional information in distance or correlation measures. In addition, in practice, only a few pairs of preferences are known for each instance instead of a total order of all possible labels. Consequently, a second line of future research is to improve the existing methods to deal with incomplete ranking information. Other interesting opportunities for future label ranking research will be the adaption towards other available classification algorithms.

## ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their insightful comments that helped to considerably improve this paper. This work was supported by the Ningbo Natural Science Foundation of China (GrantNo.2011A610177) and partially supported by the Zhejiang Provincial Natural Science Foundation of China (GrantNo.Y1101202).

## REFERENCES

[1] O. Dekel, Y. Singer, and C. D. Manning, "Log-linear models for label ranking," in *Advances in Neural Information Processing Systems*, 2003.

[2] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker, "Label ranking by learning pairwise preferences," *Artif. Intell.*, vol. 172, no. 16-17, pp. 1897–1916, 2008.

[3] A. Aiguzhinov, C. Soares, and A. P. Serra, "A similarity-based adaptation of naive bayes for label ranking: Application to the metalearning problem of algorithm recommendation," in *Discovery Science*, 2010, pp. 16–26.

[4] T. J. Hestilow and Y. Huang, "Clustering of gene expression data based on shape similarity," *EURASIP J. Bioinformatics Syst. Biol.*, vol. 2009, pp. 3:1–3:12, Jan. 2009.

[5] R. Zeng and Y.-y. Wang, "Research of personalized web-based intelligent collaborative learning," *Journal of Software*, vol. 7, no. 4, pp. 904–912, 2012.

[6] P. B. Brazdil, C. Soares, and J. P. da Costa, "Ranking learning algorithms: Using IBL and meta learning on accuracy and time results," *Machine Learning*, vol. 50, no. 3, pp. 251–277, 2003.

[7] P. L. H. Yu, W. M. Wan, and P. H. Lee, *Decision Tree Modeling for Ranking Data*, 2011, p. 83.

[8] W. Cheng and E. Hüllermeier, "Instance-based label ranking using the mallows model," in *ECCBR Workshops*, 2008, pp. 143–157.

[9] W. Cheng, K. Dembczynski, and E. Hüllermeier, "Label ranking methods based on the plackett-luce model," in *ICML*, 2010, pp. 215–222.

[10] M. Gurrieri, X. Siebert, P. Fortemps, S. Greco, and R. Slowinski, "Label ranking: A new rule-based label ranking method," in *IPMU (1)*, 2012, pp. 613–623.

[11] T. Gärtner and S. Vembu, "Label Ranking Algorithms: A Survey," in *Preference Learning*, E. H. Johannes Fürnkranz, Ed. Springer–Verlag, 2010.

[12] G. Ribeiro, W. Duivesteijn, C. Soares, and A. J. Knobbe, "Multilayer perceptron for label ranking," in *ICANN (2)*, 2012, pp. 25–32.

[13] C. R. de Sá, C. Soares, A. M. Jorge, P. J. Azevedo, and J. P. da Costa, "Mining association rules for label ranking," in *PAKDD (2)*, 2011, pp. 432–443.

[14] M. Grbovic, N. Djuric, and S. Vucetic, "Learning from pairwise preference data using gaussian mixture model," *Preference Learning: Problems and Applicaitons in AI*, p. 33, 2012.

[15] J. Fürnkranz and E. Hüllermeier, "Pairwise preference learning and ranking," in *ECML*, 2003, pp. 145–156.

[16] W. Cheng and E. Hüllermeier, "Probability estimation for multi-class classification based on label ranking," in *ECML/PKDD (2)*, 2012, pp. 83–98.

[17] C. Spearman, "The proof and measurement of association between two things." *The American journal of psychology*, vol. 15, pp. 72–101, 1904.

[18] M. G. Kendall, *Rank Correlation Methods*. London, England: Griffin, 1970.

[19] R. Kumar and S. Vassilvitskii, "Generalized distances between rankings," in *Proceedings of the 19th international conference on World wide web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 571–580.

[20] S. Har-peled, D. Roth, and D. Zimak, "Constraint classification for multiclass classification and ranking," in *In Proceedings of the 16th Annual Conference on Neural Information Processing Systems, NIPS-02*. MIT Press, 2003, pp. 785–792.

[21] J. Fürnkranz and E. Hüllermeier, "Preference learning and ranking by pairwise comparison," in *Preference Learning*. Springer-Verlag, 2010, pp. 65–82.

[22] 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*. ACM, 2009, pp. 161–168.

[23] W. Cheng and E. Hüllermeier, "A new instance-based label ranking approach using the mallows model," in *ISNN (1)*, 2009, pp. 707–716.

[24] B. Schölkopf and A. J. Smola, *Learning with kernels*. The MIT Press, 2002.

[25] J. Fürnkranz, "Round robin classification," *The Journal of Machine Learning Research*, vol. 2, pp. 721–747, 2002.

[26] E. Hüllermeier and J. Fürnkranz, "On predictive accuracy and risk minimization in pairwise label ranking," *J. Comput. Syst. Sci.*, vol. 76, no. 1, pp. 49–62, 2010.

[27] C. L. Mallows, "Non-null ranking models." *Biometrika*, vol. 44, no. 1-2, pp. 114–130, June 1957.

[28] R. D. Luce, *Individual Choice Behavior a Theoretical Analysis*. John Wiley and sons, 1959.

[29] R. Plackett, "The analysis of permutations," *Applied Statistics*, pp. 193–202, 1975.

[30] T. Qin, X. Geng, and T.-Y. Liu, "A new probabilistic model for rank aggregation," in *NIPS*, 2010, pp. 1948–1956.

[31] D. R. Hunter, "MM Algorithms for Generalized Bradley-Terry Models," *The Annals of Statistics*, vol. 32, no. 1, pp. 384–406, 2004.
[32] L. Bottou, "Stochastic learning," in *Advanced Lectures on Machine Learning*, 2003, pp. 146–168.
[33] M. Vogt, J. Godden, and J. Bajorath, "Bayesian interpretation of a distance function for navigating high-dimensional descriptor spaces," *Journal of chemical information and modeling*, vol. 47, no. 1, pp. 39–46, 2007.
[34] T. Dong, W. Shang, and H. Zhu, "An improved algorithm of bayesian text categorization," *Journal of Software*, vol. 6, no. 9, pp. 1837–1843, 2011.
[35] W. Shu and L. Ding, "Ecoga: Efficient data mining approach for fuzzy association rules," *Journal of Software*, vol. 6, no. 1, pp. 91–99, 2011.
[36] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *VLDB*, 1994, pp. 487–499.

**Yangming Zhou** is currently an M.S. student in Department of Control Science and Engineering at Zhejiang University, Hangzhou, China. His current research interest includes machine learning and data mining.

**Yangguang Liu** is currently an associate professor at Ningbo Institute of Technology, Zhejiang University, China. His current research interest includes machine learning and its applications, and artificial intelligence.

**Jiangang Yang** is currently an professor at College of Computer science and Technology, Zhejiang University, China. His current research interest includes advanced computing, and artificial intelligence.

**Xiaoqi He** is currently a lecturer at Ningbo Institute of Technology, Zhejiang University, China. His current research interest includes machine learning, and network security.

**Liangliang Liu** is currently an M.S. student in College of Computer Science and Technology at Zhejiang University, Hangzhou, China. His current research interest includes machine learning and data mining.