Label Distribution Learning by Maintaining Label Ranking Relation

Xiuyi Jia, *Member, IEEE*, Xiaoxia Shen, Weiwei Li, Yunan Lu, and Jihua Zhu, *Member, IEEE*

Abstract—Label distribution learning (LDL) is a novel machine learning paradigm that can be seen as an extension of multi-label learning (MLL). Compared with MLL, the advantages of LDL are reflected in the following perspectives: (1) the label distribution gives the relevance description of each label to unknown instances in quantitative terms; (2) the distribution implicitly gives the relevance intensities relation of different labels to a particular instance in qualitative terms, i.e., the label ranking relation. All existing LDL models aim to fit the ground-truth label distribution by quantitatively minimizing the distance between distributions or maximizing the similarity between distributions, which only uses the first advantage of the label distribution but ignores the label ranking relation, which may lose some useful semantic information implied in the label distribution, thus reducing the performance of LDL. Therefore, we propose a novel algorithm to solve this problem by introducing the ranking loss function to LDL. In addition, in order to evaluate the LDL algorithms more comprehensively and verify that the ranking loss is beneficial for keeping the label ranking relation, we also introduce two popular ranking evaluation metrics for LDL. The experimental results on 13 real-world datasets validate the effectiveness of our method.

Index Terms—Label distribution learning, label ranking, multi-label learning, learning with ambiguity

1 Introduction

Learning with ambiguity is a hot topic in recent machine learning and data mining research. There are currently two sophisticated paradigms for solving label ambiguity, namely, Single-Label Learning (SLL) and Multi-Label Learning (MLL) [1], respectively. However, both SLL and MLL are concerned only with which label or labels are related to the instance but cannot give a specific relevance degree. The Label Distribution Learning (LDL) paradigm proposed by Geng [2] can describe the relative importance of each label to a particular instance. Compared with SLL and MLL, LDL has richer semantic information and can better represent the label ambiguity. Both SLL and MLL can be regarded as special cases of LDL. Fig. 1 shows the three learning paradigms based on the LDL framework.

More researchers have applied LDL to solve the problem of label ambiguity and achieved good results. In terms of the loss function, all existing works attempt to minimize the distribution distance or maximize the distribution similarity, whether in the original space or the mapping space. For example, in the original data space, both LDLLC [3] and SLDL [4] used K-L divergence to minimize the distribution distance, while DLDL-v2 [5] adopted L1 loss. In other works, MSLP [6] looked for an optimal embedding space, and BC-LDL [7] introduced a binary coding space, then they all found the neighbor relations of unknown instances for prediction in the corresponding mapping space.

LDL describes an instance by a label distribution, as shown in Fig. 2, which not only gives the relevance degree of each label to the instance in quantitative terms but also implies the relevance intensities relation of different labels to the instance in qualitative terms, i.e., the label ranking relation. The label ranking relation is one of the important pieces of information contained in the label distribution, especially in some real-world applications that tend to focus on the top few labels. For example, a movie may simultaneously involve many different labels such as drama, love, disaster, classic and Oscar, but the recommendation website only shows the first three categories.

Label ranking relation is a qualitative representation of label distribution. If a learner can predict a "perfect" label distribution, the result with "exact" label description degrees will naturally contain the true ranking relation of labels, but unfortunately, all existing LDL methods cannot guarantee that the learner predicts fully accurate results. In the existing LDL framework, the objective loss functions and evaluation metrics can only minimize the distribution distance or maximize the distribution similarity from a quantitative perspective, but cannot extract and evaluate the qualitative characteristics of the label distribution.

The main shortcoming of the existing LDL framework is the incompleteness of the evaluation system. The current evaluation metrics are based on the distribution distance/similarity only, which can be seen from the definitions of these metrics in the experimental section (Table 2). For the predicted distribution and the ground-truth distribution,

- *Xiuyi Jia is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China, and also with the Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China. E-mail: jiaxy@njust.edu.cn.*
- *Xiaoxia Shen and Yunan Lu are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China. E-mail: {xiaoxiashen, luyn}@njust.edu.cn.*
- *Weiwei Li is with the Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China. E-mail: liweiwei@nuaa.edu.cn.*
- *Jihua Zhu is with the School of Software Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China. E-mail: zhujh@xjtu.edu.cn.*

Manuscript received 24 Nov. 2020; revised 6 May 2021; accepted 16 July 2021.
Date of publication 26 July 2021; date of current version 10 Jan. 2023.
(Corresponding author: Weiwei Li.)
Recommended for acceptance by B. Moseley.
Digital Object Identifier no. 10.1109/TKDE.2021.3099294

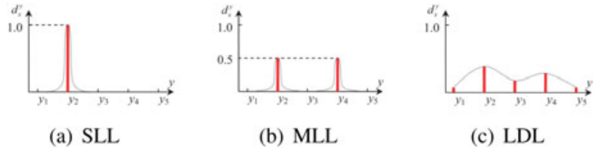


Fig. 1. The three learning paradigms based on the LDL framework [2].

each distance/similarity based measure considers the difference of each corresponding label and sums these differences in different ways, which implicitly assumes that the labels are independent of each other. However, the existence of label correlation is an inherent characteristic of the LDL, and existing evaluation metrics cannot measure the qualitative relationship between labels. As shown in Fig. 3, two predicted label distributions in Figs. 3a and 3b have the same K-L divergence. But it is obvious that the prediction in Fig. 3b is better than the prediction in Fig. 3a because the latter violates the ranking relation between labels. Furthermore, most existing LDL methods adopt quantitative metrics represented by K-L divergence as the loss function and are unable to extract the qualitative relationship between labels.

Therefore, intuitively, we can introduce label ranking loss functions into LDL to achieve the result shown in Fig. 3b. Ranking algorithms, such as McRank [8], RankBoost [9], and NDCG [10], have been widely used in information retrieval, natural language processing, data mining and other fields. Unfortunately, none of these methods can be directly applied to LDL. Specifically, unlike previous research on learning to rank problems, the label ranking in LDL needs to address not only the qualitative ranking relation between labels but also the quantitative description degree of each label.

To solve the above problems, we propose a novel LDL method by maintaining label ranking relation in this paper, namely, LDL-LRR. First, we propose a novel ranking loss function for label distribution characteristics, and combine it with K-L divergence as the loss term of the objective function, which not only guarantees the trained model to fit the ground-truth label distribution as much as possible but also maintains the label ranking relation within a certain error range. At the same time, considering the true pairwise labels with different spacings, the larger the distance, the more serious the influence will be, so we introduce the weight coefficient to penalize the different error rankings. In addition, existing LDL evaluation metrics are all based on distance or similarity to measure the performance of the algorithm, which only considers the quantitative semantic of the label distribution. Therefore, in the experiment, besides the traditional metrics that evaluate the similarity or distance of any two distributions, we also evaluate their ability of maintaining the ranking relation. Finally, we implement comparison experiments on 13 datasets, and the results show that our proposed method can not only fit the value of the label distribution well, but also greatly improve the ability of maintaining the label ranking relation.

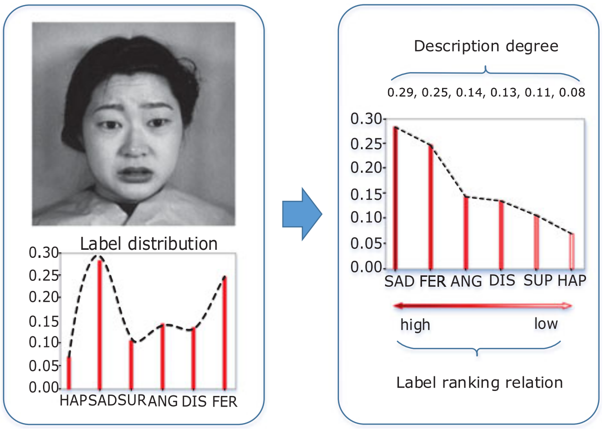


Fig. 2. Illustration of the fact that the label distribution contains two semantics: the relevance degree of each label to the instance and the relevance intensities relation of different labels to the instance, i.e., the label ranking relation.

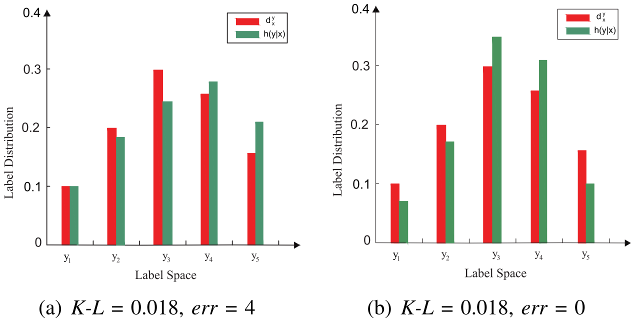


Fig. 3. An example to illustration of the fact that the loss function used by the previous LDL algorithms cannot maintain the label ranking relation. Red bars represent the ground-truth label distribution and green bars represent the predicted label distribution. The *K-L* is the value of K-L divergence and the *err* is the number of error ranking between labels.

The main contribution of this paper is in three aspects:

- We propose a novel method, LDL-LRR, to address LDL models by considering label ranking relation.
- We design a novel ranking loss function for LDL and combine it with K-L divergence as the loss term of the objective function, which can guarantee that the trained model fits the ground-truth label distribution from both qualitative and quantitative perspectives.
- We expand the LDL experiment. Specifically, in addition to the conventional evaluation metrics of distance or similarity, we also propose that the algorithm performance should be measured from the perspective of label ranking relation.

The remainder of the paper is organized as follows. In Section 2, we briefly discuss existing works related to LDL and learning to rank problem. Our proposed method will be presented in Section 3. Finally, we report the experimental results in Section 4, and followed by the conclusion in Section 5.

2 Related Work

2.1 Label Distribution Learning

To solve the problem of insufficient training samples in the age estimation problem, Geng *et al.* [11] first proposed an algorithm IIS-LLD based on the idea of label distribution. Later, Geng [2] proposed LDL as a complete learning framework and formally gave the definition of LDL.

After LDL was proposed, an increasing number of researchers used it to solve the problem of label ambiguity and achieved good results. In summary, existing LDL algorithms can be divided into three categories: problem transformation (PT), algorithm adaptation (AA) and specialized algorithms (SA). Problem transformation algorithms transform the LDL problem into the traditional problem and then use existing learners to solve it, such as PT-SVM and PT-Bayes [2]. The main idea of algorithm adaptation is to adapt traditional algorithms to fit LDL paradigm, such as AA-kNN [2] and LogitBoost [12]. Different from the indirect strategy of problem transformation and algorithm adaptation, the specialized algorithms directly match the LDL problem. For example, LDL-SCL [13] directly models the relative importance of each label to a particular instance. Relevant studies and experiments have shown that the algorithms based on the third design strategy have better results than the algorithms based on the first two design strategies.

Generally speaking, a specialized algorithm consists of three parts, i.e., an output model, an objective function and an optimization method. For the output model, the specialized LDL algorithms mainly use the maximum entropy model [14], such as the BFGS-LLD algorithm [2] and the EDL_LRL algorithm [15]. For the objective function, it usually chooses some functions that can measure the distance or similarity between two distributions, that aims to fit the label distribution values. Many divergence functions are adopted in existing research, such as the K-L (Kullback-Leibler) divergence in the IIS-ALDL algorithm [16], the balanced divergence function defined in the EDL algorithm [17], Jeffery divergence [18], and weighted Jeffery divergence [19]. Among them, K-L divergence is widely used. To solve the optimization problem, many methods are also applied in LDL. EDL [17] uses L-BFGS method as the optimization method. LDLSP algorithm [20] uses ADMM (Alternating Direction Method of Multipliers) as the optimization method. LDL4C [21] uses BFGS method.

Similarly, our specialized LDL algorithm is also constructed from above three aspects. In this paper, we introduce a novel ranking loss function to maintain the label ranking relation for LDL methods. To specify the importance of label ranking relation in the objective function, both the output model and the optimization method are the general approach in the LDL field.

2.2 Learning to Rank

Learning to rank is a kind of machine learning technique used to solve the ranking problem. It is first introduced for document retrieval. Specifically, given a query, the document retrieval system first finds documents containing query words, then learning to rank models sort these documents, and return the top-ranked documents to the user.

Existing learning to rank algorithms mainly fall into three major categories. The first is called pointwise based approach, where each document is treated as a single instance and their relevance to the query is considered as the label. Prominent examples are SubNet Ranking [22], OCSVM [23] and McRank [24]. They use some traditional machine learning techniques to tackle ranking problem. Note that these methods are restricted to linear orders but cannot capture more general relations, such as partial orders.

The second is called pairwise based approach, where the instance is the pair of documents and the label is the relative order between the two documents. Then, the ranking problem can be reduced to a classification or regression problem on the document pair. Ranking SVM [25] is one of the typical pairwise based methods, which uses the differences between feature vectors of objects to train a SVM. Later, RankNet [26] is proposed to train a multilayer perceptron (MLP). Further, some extensions of these approaches are developed. GBRank [27] and LambdaRank [28] use some boosting models instead of SVM and MLP. In addition, RankGNNs [29] addresses the problem of learning to rank graph-structured data by combining the neural pairwise ranking models and graph neural networks.

The third is called listwise based approach. It solves the ranking problem by directly minimizing the difference between predicted ranked list and the ground-truth. One of the first listwise losses is ListNet [30], which uses a probability distribution over all possible rankings of objects and uses the cross-entropy as learning criterion. Intuitively, listwise based methods outperform both pointwise and pairwise based methods, since they completely maintain the ranking relation with high time complexity. However, a generalization [31] of RankNet is proposed to show that pairwise methods can still be competitive with the more recent and much more complex listwise based methods while requiring much shorter training time.

Another area related to the ranking problem is rank aggregation [32]. They focus on how to aggregate a number of candidate ranking results [33], [34], yet we focus on how to measure the difference between two ranking results. By considering both the performance and the time complexity, we adopt the ranking loss function from pairwise based methods to maintain the label ranking relation in this paper.

3 The Proposed Method

3.1 Problem Formulation

Let $\mathcal{X} = [x_1; x_2; \dots; x_n] \in \mathcal{R}^{n\times q}$ denote the input space, where x_i is the ith instance, n is the number of instances and q is the dimension of features. Let $\mathcal{D} = [\mathcal{D}_1; \mathcal{D}_2; \dots; \mathcal{D}_n] \in \mathcal{R}^{n\times l}$ denote the output space, where $\mathcal{D}_i = [d_i^1, d_i^2, \dots, d_i^l]$ is the label distribution associated with instance x_i, d_i^j is used to indicate the importance of label y_j to instance x_i, which satisfies $d_i^j \in [0,1]$ and $\sum_{j=1}^l d_i^j = 1$, and l is the number of labels. Given a training set $S = \{(x_1, \mathcal{D}_1), (x_2, \mathcal{D}_2), \dots, (x_n, \mathcal{D}_n)\}$, the goal of LDL is to learn a mapping function $f: \mathcal{X} \rightarrow \mathcal{D}$ from S that can predict the label distributions for unseen instances.

3.2 LDL-LRR

Suppose that $h(y|x;\theta)$ is the output model learnt from S, where θ is the parameter matrix. The goal of LDL is to find an appropriate θ that can generate a distribution y_i similar to $\mathcal{D}_i$ given an instance x_i. Moreover, as for the form of $h(y|x;\theta)$, we assume it to be a maximum entropy model similar to previous work [11] as follows:

$$h(y_j|x_i;\theta) = \frac{1}{Z_i}\exp\bigg(\sum_k x_i^k \theta_{k,j}\bigg), \tag{1}$$

where $h(y_j|x_i;\theta)$ is the predicted label description degree of label y_j to instance x_i, $\theta_{k,j}$ is an element in the parameter matrix θ, x_i^k is the kth feature of x_i, and $Z_i = \sum_j \exp$

$(\sum_k x_i^k \theta_{k,j})$ is a normalization term to satisfy the sum of all label description degrees of an instance equals to 1. To simplify the formula, we use $h(y_j|x_i)$ to represent $h(y_j|x_i;\theta)$. In addition to maintaining the label ranking relation, our model also needs to fit the ground-truth label information as much as possible. Therefore, the objective function is constructed as follows:

$$\min_{\theta} V(\theta) + \lambda R(\theta) + \beta\Omega(\theta), \tag{2}$$

where V is the fitting loss term on the training set to measure the similarity or distance between the true distribution and predicted distribution, R is the ranking loss term to maintain the label ranking relation, Ω is a regularizer to control the complexity of the output model, λ and β are two trade-off parameters.

First, considering that the goal of LDL is to make the predicted distribution as similar as possible to the true distribution, the loss term V should be a function that can measure the similarity of two distributions. Cha [35] analyzed many such functions, such as K-L divergence, Jeffery divergence and K divergence. Here, we refer to the method suggested by Geng [2] in a survey of LDL and then use K-L divergence as our fitting loss term defined by

$$D_J(Q_a||Q_b) = \sum_j Q_a^j \ln\frac{Q_a^j}{Q_b^j}, \tag{3}$$

where Q_a^j and Q_b^j are the jth element of the two distributions Q_a and Q_b, respectively. Specifically, in this paper, the expression for V based on K-L divergence is defined as follows:

$$V(\theta) = \sum_{i=1}^{n}\sum_{j=1}^{l}\left(d_i^j \ln\left(\frac{d_i^j}{h(y_j|x_i)}\right)\right), \tag{4}$$

where d_i^j and $h(y_j|x_i)$ denote the true description degree and the predicted description degree of label y_j to instance x_i, respectively.

Next, we propose a ranking loss function to exploit the label ranking relation. Many ranking loss functions have been studied for the learning to rank problem. Inspired by RankNet [36], the cross-entropy is applied to our ranking loss function:

$$\begin{aligned} R(\theta) = \frac{1}{N}\sum_{i=1}^{n}\sum_{u=1}^{l}\sum_{v=1}^{l} -&((1-p_{u,v}^i)\log(1-\hat{p}_{u,v}^i) \\ &+ p_{u,v}^i \log\hat{p}_{u,v}^i) \end{aligned} \tag{5}$$

where $N = n \times l \times l$, $p_{u,v}^i$ is the true correlation probability, representing the probability that the label y_u ranks before y_v for the instance x_i in the true label space, and $\hat{p}_{u,v}^i$ is the predicted correlation probability, representing the probability that y_u ranks before y_v in the predicted label space. The true correlation probability matrix P is defined as follows:

$$P = \left[\begin{bmatrix} p_{1,1}^1 & \cdots & p_{1,l}^1 \\ \vdots & \ddots & \vdots \\ p_{l,1}^1 & \cdots & p_{l,l}^1 \end{bmatrix} \cdots \begin{bmatrix} p_{1,1}^n & \cdots & p_{1,l}^n \\ \vdots & \ddots & \vdots \\ p_{l,1}^n & \cdots & p_{l,l}^n \end{bmatrix}\right], \tag{6}$$

where $p_{u,v}^i$ is defined by

$$p_{u,v}^i = \begin{cases} 1, & \text{if } d_i^u > d_i^v \\ 1/2, & \text{if } d_i^u = d_i^v \\ 0, & \text{if } d_i^u < d_i^v \end{cases}. \tag{7}$$

Considering that the matrix $\hat{P}$ of the predicted label space should be consistent with the P of the true space, we introduce the Sigmoid function to construct a convex problem as follows:

$$\hat{p}_{u,v}^i = \frac{1}{1+\exp(-\sigma(h(y_u|x_i) - h(y_v|x_i)))}. \tag{8}$$

We can adjust the size of σ to make $\hat{p}_{u,v}^i$ approximately equal to the definition of $p_{u,v}^i$ in the true label space. In this paper, $\sigma = 100$. When $h(y_u|x_i)$ is greater than $h(y_v|x_i)$, $\hat{p}_{u,v}^i$ is very close to 1, and the opposite also holds.

When the error ranking between labels occurs, for the pairwise ground-truth labels with different spacings, the greater the distance, the stronger the punishment should be. Therefore, we use the square of the distance between two ground-truth labels as the weight coefficient to improve R. Obviously, both P and $\hat{P}$ are symmetric matrices. To avoid double counting, multiply R by 0.5. Thus, the final ranking loss function is:

$$\begin{aligned} R(\theta) = \frac{1}{2N}\sum_{i=1}^{n}\sum_{u=1}^{l}\sum_{v=1}^{l} -&((1-p_{u,v}^i)\log(1-\hat{p}_{u,v}^i) \\ &+ p_{u,v}^i \log\hat{p}_{u,v}^i)(d_i^u - d_i^v)^2. \end{aligned} \tag{9}$$

For the third term of Eq. (2), we utilize the F-norm of the matrix to implement as follows:

$$\Omega(\theta) = \frac{1}{2} \parallel \theta \parallel_F^2 . \tag{10}$$

In summary, the final objective function is as follows:

$$\begin{aligned} \min_{\theta} \frac{\lambda}{2N}\sum_{i=1}^{n}\sum_{u=1}^{l}\sum_{v=1}^{l} &-((1-p_{u,v}^i)\log(1-\hat{p}_{u,v}^i) \\ &+ p_{u,v}^i \log\hat{p}_{u,v}^i)(d_i^u - d_i^v)^2 + \frac{\beta}{2} \parallel \theta \parallel_F^2 \\ &+ \sum_{i=1}^{n}\sum_{j=1}^{l}\left(d_i^j \ln\left(\frac{d_i^j}{h(y_j|x_i)}\right)\right), \end{aligned} \tag{11}$$

Algorithm 1. The LDL-LRR Algorithm

```
Input: training set $S = \{\mathcal{X}, \mathcal{D}\}$, parameters $\lambda$ and $\beta$.
Output: the label distribution $\mathcal{D}_t$.
initialize $\theta$, $\alpha$, $\beta_{11}$, $\beta_2$, $t = 1$;
compute the true correlation probability matrix $P$ by Eqs. (6) and (7);
repeat
    compute $\nabla\theta$ by Eqs. (14) and (15);
    update $m_t$ and $v_t$ by Eq. (12);
    update $\theta$ by Eq. (13);
    $t = t + 1$;
until stopping criterion is satisfied
return the label distribution $\mathcal{D}_t$ according to Eq. (1).
```

3.3 Optimization

We adopt AMSGRAD [37] as the optimization method for our proposed model. AMSGRAD is an improvement on Adam, which not only stores an exponentially decaying average of past squared gradients m_t but also saves the exponential decay maximum of the past gradient square v_t and uses it to update the learning rate. The formula of m_t and v_t is as follows:

$$\begin{aligned} m_t &= \beta_{1t} m_{t-1} + (1-\beta_{1t}) g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t, \end{aligned} \tag{12}$$

where β_{1t} and β_2 are the decay rates, which are always recommended closing to 1. The quantity essentially is defined as $\Gamma_{t+1} = \left(\frac{\sqrt{V_{t+1}}}{\alpha_{t+1}} - \frac{\sqrt{V_t}}{\alpha_t} \right)$, where α_t is step size and $V_t = \mathrm{d}iag(v_t)$. In order to satisfy the quantity essentially $\Gamma_{t+1} \geq 0$, the super-parameter β_{1t} is set to change with t, i.e., $\beta_{1t} = \frac{\beta_{11}}{t}$. g_t is the gradient of objective function at the tth iteration. Then, the rule for updating parameters with AMSGRAD is as follows:

$$\Theta_{t+1} = \Theta_t - \alpha_t m_t / \sqrt{\hat{v}_t}, \tag{13}$$

where Θ denotes the parameter that needs to update, $\alpha_t = \frac{\alpha}{\sqrt{t}}$ is the learning rate, $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$.

Besides, the Mini-batch gradient descent strategy is applied to update the parameters during the training, which performs an update for every Mini-batch of training examples. Its advantages are that (1) it uses only a subset of training set in an update, thus it can speed up the update and save memory; (2) it reduces the variance of the parameter updates, which can lead to more stable convergence. Therefore, we obtain the gradient of Eq. (11), which is as follows:

$$\frac{\partial T(\theta)}{\partial \theta_{k,j}} = \begin{cases} \frac{\lambda}{2N} \sum_i \left(\sum_m \frac{(d^i_j - d^i_m)^2 (\hat{p}^i_{j,m} - p^i_{j,m})}{\hat{p}^i_{j,m}(1-\hat{p}^i_{j,m})} \frac{\partial \hat{p}^i_{j,m}}{\partial \theta_{k,j}} \right. \\ \left. + \sum_q \frac{(d^i_l - d^i_j)^2 (\hat{p}^i_{q,j} - p^i_{q,j})}{\hat{p}^i_{q,j}(1-\hat{p}^i_{q,j})} \frac{\partial \hat{p}^i_{q,j}}{\partial \theta_{k,j}} \right) \\ + \sum_i - \frac{d^j_i}{h(y_j|x_i)} \frac{\partial h(y_j|x_i)}{\partial \theta_{k,j}} + \beta \theta_{k,j} \end{cases} \tag{14}$$

where

$$\begin{aligned} \frac{\partial \hat{p}^i_{j,m}}{\partial \theta_{k,j}} &= \hat{p}^i_{j,m}(1 - \hat{p}^i_{j,m}) \frac{\partial h(y_j|x_i)}{\partial \theta_{k,j}} (-\sigma) \\ \frac{\partial \hat{p}^i_{q,j}}{\partial \theta_{k,j}} &= \hat{p}^i_{q,j}(1 - \hat{p}^i_{q,j}) \frac{\partial h(y_q|x_i)}{\partial \theta_{k,j}} \sigma \\ \frac{\partial h(y_j|x_i)}{\partial \theta_{k,j}} &= \frac{\exp(x_{i.}\theta_{.j}) x^k_i \sum_{m \neq j} \exp(x_{i.}\theta_{.m})}{\left(\sum_{m'} \exp(x_{i.}\theta_{.m'})\right)^2}. \end{aligned} \tag{15}$$

The overall procedure of the proposed LDL-LRR algorithm is shown in Algorithm 1. We obtain the optimal parameter θ through the AMSGRAD algorithm and then predict the unseen instances by Eq. (1).

3.4 Complexity Analysis

During the procedure of optimization, the main time cost is to calculate the gradient and update parameters. The complexity of computing $\nabla \theta$ and updating θ is $O(nql^2)$ and $O(ql)$, respectively. Therefore, the complexity of each iteration is $O(nql^2)$, where n, q, and l are the number of instances, features, and labels, respectively. It is clear that our algorithm can be applied to large-scale datasets.

TABLE 1
Statistics of the 13 Datasets

No.	Dataset	Examples	Features	Labels
1	s-JAFFE (sj)	213	243	6
2	Movie (mov)	7755	1869	5
3	Natural_Scene (ns)	2000	294	9
4	Human_Gene (gene)	17892	36	68
5	Emotion6 (emo)	1980	168	7
6	Yeast-alpha (alpha)	2465	18	6
7	Yeast-cdc (cdc)	2465	24	15
8	Yeast-elu (elu)	2465	24	14
9	Yeast-diau (diau)	2465	24	7
10	Yeast-heat (heat)	2465	24	6
11	Yeast-spo5 (spo5)	2465	24	3
12	Yeast-cold (cold)	2465	24	4
13	Yeast-dtt (dtt)	2465	24	4

4 Experiments

4.1 Datasets

We carry out our experiments on 13 label distribution datasets, including two facial expression datasets, s-JAFFE [38] and emotion6 [39]; eight biological experiment datasets, Yeast [40]; a natural scene dataset, Natural_Scene [41]; a movie category dataset, Movie; and a large-scale biomedical research dataset, Human Gene.[1]

The characteristics of the 13 datasets are summarized in Table 1.

In detail, the dataset s-JAFFE is a widely used facial expression image dataset, which is the extensions of JAFFE. There are 213 gray scale expression images posed by 10 Japanese female models, and each image is scored by 60 persons on the 6 basic emotion labels (i.e., happiness, sadness, surprise, fear, anger and disgust) with a five-level scale (1 represents the lowest emotion intensity, while 5 represents the highest emotion intensity). Different from most works on JAFFE, the average scores (after normalization) are used to represent the label distribution, rather than only considering the emotion with the highest score as a single-label problem, and the data set is extended to s-JAFFE (Scored JAFFE).

The dataset Movie is about the user rating on movies coming from Netflix. There are 7,755 movies and 54,242,292 ratings from 478,656 users. For each movie, it is scored on a scale from 1 to 5 integral stars. The percentage of each rating level is calculated as the label distribution for each image. Besides, the features of a movie are extracted from the meta data (i.e., genre, director, actor, country, etc.), and the feature vector is of 1,869 dimensions by attribute transformation.

The dataset Nature_Scene is collected from 2,000 natural scene images with inconsistent multi-label ranking. Ten human rankers are demanded to label these images with nine possible labels, i.e., plant, sky, cloud, snow, building,

[1] These datasets can be downloaded from http://ldl.herokuapp.com/download

TABLE 2
Evaluation Metrics for LDL Algorithms

	Name	Formula
Distance	*Chebyshev* ↓	$Dis_1(\bar{D}, D) = \max_j \lvert \bar{d}_j - d_j \rvert$
	Clark ↓	$Dis_2(\bar{D}, D) = \sqrt{\sum_{j=1}^{L} \frac{(\bar{d}_j - d_j)^2}{(\bar{d}_j + d_j)^2}}$
	K-L ↓	$Dis_3(\bar{D}, D) = \sum_{j=1}^{L} \bar{d}_j \ln \frac{\bar{d}_j}{d_j}$
	Canberra ↓	$Dis_4(\bar{D}, D) = \sum_{j=1}^{L} \frac{\lvert \bar{d}_j - d_j \rvert}{\bar{d}_j + d_j}$
Similarity	*Cosine* ↑	$Sim_1(\bar{D}, D) = \frac{\sum_{j=1}^{L} \bar{d}_j d_j}{\sqrt{\sum_{j=1}^{L} \bar{d}_j^2} \sqrt{\sum_{j=1}^{L} d_j^2}}$
	Intersection ↑	$Sim_2(\bar{D}, D) = \sum_{j=1}^{L} \min(\bar{d}_j, d_j)$
Ranking	*Spearman's rank* (ρ_S) ↑	$\rho_S = 1 - \frac{6 \sum_i (\bar{D} - D)^2}{k(k^2 - 1)}$
	Kendall tau correlation coefficient (τ_K) ↑	$\tau_K = \frac{n_c - n_d}{\frac{1}{2} k(k-1)}$

$D = \{d_1, d_2, \ldots, d_L\}$ denotes the predicted label distribution and $\bar{D} = \{\bar{d}_1, \bar{d}_2, \ldots, \bar{d}_L\}$ denotes the real label distribution. ↑ (↓) indicates the higher (lower), the better.

desert, mountain, water and sun. For each image, each human ranker selects the relevant labels and ranks them in descending order independently. Thus, the result of multi-label rankings is expected to be highly inconsistent. Then, a non-linear programming process is employed to transform the inconsistent rankings into a label distribution [41]. Finally, with the method proposed in [42], a 294-dimensional feature vector is extracted for each image.

The dataset Human_Gene is a large-scale data set with 17,892 examples, which is collected from the biological research that studies the relation between human genes and diseases. Each of example has 36 features that denote 36 numerical descriptors for a sequence and 68 labels that denote 68 different diseases. Then, the gene expression levels after normalization constitute the label distribution of a particular human gene.

Emotion 6 is assembled from Flickr for a sentiment prediction benchmark, which is annotated with the votes for seven emotional categories (i.e., anger, disgust, joy, fear, sadness, surprise and neutral), containing a total of 1980 images.

The last eight datasets used in the experiments are collected from biological experiments on the budding yeast Saccharomyces cerevisiae, where each data set denotes the result of an experiment. There are 2465 yeast genes in total, each of which is described by an associated phylogenetic profile vector with length 24. The normalized description degree of the corresponding label represents the gene expression level in different time points.

4.2 Evaluation Metrics

In this paper, eight metrics, including six distance-based or similarity-based metrics [2] and two ranking-based metrics, are chosen as the evaluation metrics for the LDL algorithms. The names and formulas are presented in Table 2, where $D = \{d_1, d_2, \ldots, d_L\}$ denotes the predicted label distribution and $\bar{D} = \{\bar{d}_1, \bar{d}_2, \ldots, \bar{d}_L\}$ denotes the real label distribution. For the first four distance metrics, "↓" indicates "the smaller, the better", and "↑" indicates "the larger, the better" for the next two similarity metrics. The last two metrics named *spearman's rank* (ρ_S) [43] and *Kendall tau correlation coefficient* (τ_K) [44], which are popular and widely accepted similarity metrics for rankings, are used to measure the degree of ranking relation maintenance in experiments. *Spearman's rank* is a non-parametric metric of correlation between two variables. For a pair of rankings D and $\bar{D}$ of length k, it is defined as

$$\rho_S = 1 - \frac{6 \sum_i (\bar{D} - D)^2}{k(k^2 - 1)}. \tag{16}$$

Kendall tau correlation coefficient is a non-parametric statistic used to measure the degree of correspondence between two rankings, which is defined as

$$\tau_K = \frac{n_c - n_d}{\frac{1}{2} k(k - 1)}, \tag{17}$$

where $\frac{1}{2} k(k - 1)$ is the number of possible pairwise combinations. The values of this coefficient range from $[-1, 1]$, where $\tau_K(D, D) = 1$ if the rankings are equal and $\tau_K(D, D^{-1}) = -1$ if D^{-1} denotes the inverse order of D (e.g. $D = (0.1, 0.2, 0.3, 0.4)$ and $D^{-1} = (0.4, 0.3, 0.2, 0.1)$).

4.3 Experimental Setting

The proposed LDL-LRR algorithm is compared with nine state-of-the-art algorithms: BFGS-LLD [2], cos-LDL [45], LOLAT [46], LDLSF [20], LDL-SCL [13], MSLP [6], PT-Bayes [2], AA-kNN [2], CPNN [47]. The parameter settings of all algorithms are as follows. For BFGS-LLD and cos-LDL, all parameters are set as recommended in their literature, respectively. For LALOT, the trade-off parameter C and the entropic regularization coefficient λ are tuned by 10-fold cross-validation. For LDLSF, the parameters λ_1, λ_2 and λ_3 are selected from $10^{\{-6,-5,\ldots,-2,-1\}}$, respectively, and ρ is set as 10^{-3}. For LDL-SCL, default parameter values in the corresponding literature are applied. For MSLP, the parameters are set according to the requirements of parameter setting in [6]. For PT-Bayes, maximum likelihood estimation is employed to estimate the Gaussian class-conditional probability density functions. The number of neighbors k in AA-kNN is set to 5. CPNN tries to use a three layer neural network to learn the label distribution. For LDL-LRR, the parameters λ and β are selected from $10^{\{-6,-5,\ldots,-2,-1\}}$ and $10^{\{-3,-2,\ldots,1,2\}}$, respectively. The initialization of other variables is all-zero.

4.4 Results and Discussion

Comparison With State-of-the-Arts. Tables 3 and 4 report the detailed experimental results of nine comparison algorithms on all datasets, where the best performance among the comparison algorithms on each metric is marked in bold. On each dataset, 10 times 5-fold cross-validation is conducted, and the mean value and the standard deviation of each evaluation metric are recorded.

From these tables, it can be seen that the LDL-LRR algorithm achieved 100 firsts and 4 seconds on the mean value in 104 sets of results corresponding to 13 datasets and 8 evaluation metrics. In particular, it achieved the best results on all two ranking metrics shown in the last two columns of Tables 3 and 4.

To perform comparative analysis in more well-founded ways, we implement a pairwise two-tailed t-test with 0.05

TABLE 3
Comparison Results on Seven Datasets are Shown as “mean±std”

data	algorithm	*K-L*↓	*Chebyshev*↓	*Intersection*↑	*Cosine*↑	*Clark*↓	*Canberra*↓	τ_K ↑	ρ_S ↑
sj	LDL-LRR	**.0376±.0027**	**.0817±.0060**	**.8910±.0053**	**.9647±.0032**	**.3133±.0105**	**.6421±.0286**	**.6678±.0117**	**.5568±.0317**
	BFGS-LLD	.0633±.0070	.1103±.0063	.8614±.0100	.9400±.0064	.4002±.0294	.8152±.0644	.1478±.1171	.1842±.1397
	cos-LDL	.0726±.0074	.1189±.0082	.8480±.0096	.9317±.0070	.4271±.0262	.8942±.0549	.0507±.0738	.0853±.0970
	LALOT	.0873±.0087	.1282±.0081	.8358±.0085	.9178±.0080	.4514±.0199	.9452±.0448	.0704±.0457	.0827±.0500
	LDLSF	.0507±.0029	.0872±.0016	.8848±.0014	.9597±.0011	.3614±.0048	.6831±.0578	.4968±.0130	.4848±.0137
	LDL-SCL	7.8589±3.8762	.6938±.2787	.2982±.2629	.4839±.2142	1.9863±.7312	4.7377±1.9635	.0109±.1209	.0142±.1246
	MSLP	.0661±.0059	.1144±.0069	.8554±.0083	.9370±.0054	.4020±.0172	.8862±.0033	.3967±.0053	.4668±.0063
	PT-Bayes	.0738±.0003	.1204±.0004	.8466±.0004	.9304±.0003	.4293±.0012	.7549±.0057	-.0131±.0080	-.0235±.0000
	AA-kNN	.0525±.0221	.0951±.0166	.8819±.0162	.9507±.0175	.3346±.0379	.7771±.0057	.3552±.0897	.4663±.1040
	CPNN	.0594±.0037	.1068±.0020	.8631±.0045	.9433±.0032	.3856±.0110	.8883±.0601	.2128±.0523	.2228±.0601
mov	LDL-LRR	**.0982±.0031**	**.1151±.0016**	**.8355±.0023**	**.9353±.0020**	**.5235±.0065**	**.9996±.0093**	**.7053±.0063**	**.7070±.0093**
	BFGS-LLD	.1378±.0373	.1397±.0223	.8053±.0263	.9108±.0227	.5856±.0494	1.1309±.1098	.5658±.0294	.6611±.0284
	cos-LDL	.1870±.0497	.1734±.0283	.7689±.0298	.8775±.0267	.6708±.0613	1.2934±.1273	.0850±.0385	.0874±.0430
	LALOT	3.7216±.9542	.2854±.0903	.6015±.2114	.7021±.1415	1.431±.9006	2.9023±.8996	-.0423±.2317	-.0043±.3118
	LDLSF	.1954±.0257	.1271±.0027	.8180±.0034	.9220±.0028	.6227±.0112	1.0166±.0158	.6504±.0092	.6604±.0082
	LDL-SCL	.2280±.0223	.1292±.0026	.8174±.0032	.9194±.0027	.5622±.0090	1.0143±.0124	.6601±.0091	.6676±.0090
	MSLP	.1151±.0042	.1235±.0023	.8209±.0036	.9239±.0026	.5578±.0103	1.0118±.0010	.6050±.0109	.6453±.0099
	PT-Bayes	.5240±.0175	.2010±.0011	.7229±.0010	.8496±.0008	.8052±.0026	1.1057±.0212	-.0002±.0000	-.0006±.0000
	AA-kNN	.1163±.0050	.1231±.0025	.8231±.0031	.9230±.0031	.5346±.0086	1.1178±.0125	.6048±.0122	.6448±.0125
	CPNN	.1388±.0099	.1347±.0057	.8083±.0078	.9100±.0067	.5845±.0180	1.2110±.0244	.5160±.0144	.6169±.0244
ns	LDL-LRR	**.7314±.0070**	**.2997±.0070**	**.5670±.0022**	**.7532±.0044**	2.4268±.0106	**5.7143±.0126**	**.6001±.0066**	**.5301±.0084**
	BFGS-LLD	.9067±.0556	.3511±.0158	.4778±.0196	.6765±.0211	2.4756±.0217	5.8445±.0932	.3430±.0200	.4212±.0241
	cos-LDL	.9558±.0567	.3635±.0163	.4617±.0238	.6593±.0206	2.4821±.0184	5.8628±.0762	.1085±.0541	.1259±.0646
	LALOT	1.2529±.0145	.3873±.0033	.3600±.0029	.5473±.0046	2.4999±.0030	5.9962±.0251	-.0766±.0265	-.0909±.0356
	LDLSF	.9981±.0328	.3196±.0092	.5397±.0074	.7297±.0074	2.4622±.0159	6.6739±.0139	.4883±.0132	.4903±.0139
	LDL-SCL	.8908±.0135	.3388±.0095	.4824±.0058	.6951±.0063	2.4726±.0122	6.7882±.0128	.4212±.0120	.4167±.0128
	MSLP	.9900±.0258	.3623±.0102	.4641±.0069	.6495±.0082	2.4270±.0164	6.7865±.0156	.4036±.0166	.3836±.0156
	PT-Bayes	1.9070±.0222	.4088±.0015	.3490±.0008	.5580±.0008	2.5241±.0015	6.7529±.0001	-.0011±.0011	-.0025±.0001
	AA-kNN	1.1843±.0964	.3157±.0114	.5656±.0159	.7049±.0175	**1.8143±.0308**	6.5579±.0252	.4911±.0202	.4912±.0252
	CPNN	.9211±.0269	.3145±.0051	.4867±.0112	.6760±.0074	2.4595±.0054	6.8121±.0577	.4200±.0502	.3273±.0577
gene	LDL-LRR	**.2365±.0049**	**.0532±.0011**	**.7844±.0014**	**.8346±.0020**	**2.1114±.0122**	**13.5681±.0025**	**.1808±.0055**	**.1618±.0025**
	BFGS-LLD	.2398±.0038	.0539±.0009	.7828±.0014	.8328±.0018	2.1270±.0141	14.5633±.1107	.0928±.0388	.1332±.0561
	cos-LDL	.2411±.0099	.0537±.0007	.7807±.0091	.8311±.0081	2.1374±.0665	14.6700±.5110	.1489±.0832	.1532±.0979
	LALOT	.2956±.0059	.0573±.0012	.7578±.0016	.8055±.0024	4.0703±3.6917	17.8198±3.7167	.0271±.0019	.0396±.0029
	LDLSF	.2395±.0054	.0533±.0009	.7828±.0022	.8332±.0028	2.1295±.0209	14.5681±.0055	.1100±.0098	.1190±.0055
	LDL-SCL	.2372±.0043	.0536±.0011	.7840±.0012	.8338±.0018	2.1154±.0104	14.4336±.0029	.1303±.0050	.1503±.0029
	MSLP	.2532±.0044	.0558±.0011	.7707±.0014	.8168±.0021	2.2056±.0127	14.4381±.0057	.1002±.0050	.1047±.0057
	PT-Bayes	1.5543±.0514	.2027±.0131	.4609±.0138	.4363±.0237	4.7358±.0945	14.9696±.0040	.0070±.0010	.0035±.0040
	AA-kNN	.2973±.0114	.0642±.0020	.7430±.0034	.7695±.0042	2.3762±.0230	14.8304±.0042	.0886±.0022	.0972±.0042
	CPNN	.2390±.0072	.0535±.0011	.7828±.0024	.8327±.0023	2.1289±.0215	15.2365±.0179	.0077±.0109	.0089±.0179
emo	LDL-LRR	**.5966±.0136**	.3109±.0064	**.5802±.0040**	.7086±.0040	1.6599±.0154	**.6819±.0088**	**.3523±.0067**	**.3544±.0088**
	BFGS-LLD	.6260±.0101	.3172±.0055	.5719±.0030	.6954±.0066	1.6905±.0136	.8141±.0396	.2789±.0149	.3382±.0182
	cos-LDL	.6601±.0214	.3343±.0048	.5513±.0072	.6721±.0116	1.6698±.0100	.7804±.0333	.1154±.0339	.1411±.0419
	LALOT	.7216±.0041	.3518±.0013	.5234±.0014	.6447±.0015	1.7065±.0206	.8623±.0196	.0258±.0007	.0319±.0088
	LDLSF	.6040±.0044	**.3060±.0005**	.5779±.0008	**.7184±.0011**	1.6640±.0009	.6901±.0030	.3304±.0010	.3424±.0030
	LDL-SCL	.6102±.0204	.3255±.0079	.5652±.0066	.7073±.0090	**1.6431±.0184**	.6968±.0210	.3422±.0290	.3493±.0210
	MSLP	.6113±.0251	.3192±.0077	.5716±.0090	.7009±.0121	1.6609±.0190	.6970±.0245	.3009±.0190	.3199±.0245
	PT-Bayes	6.3771±.0362	.6865±.0041	.2820±.0038	.4169±.0065	2.4444±.0013	.6996±.0049	.2662±.0056	.2822±.0049
	AA-kNN	.8573±.0656	.3325±.0086	.5512±.0088	.6555±.0120	1.7167±.0214	.7028±.0267	.2770±.0200	.2870±.0267
	CPNN	1.4319±.0846	.4179±.0142	.4460±.0112	.5216±.0155	1.8898±.0190	.7519±.0115	.1211±.0123	.1741±.0115
alpha	LDL-LRR	**.0054±.0001**	**.0134±.0002**	**.9625±.0005**	**.9946±.0001**	**.2093±.0031**	**.6791±.0100**	**.2058±.0158**	**.2144±.0100**
	BFGS-LLD	.0055±.0001	.0135±.0001	.9629±.0006	.9949±.0001	.2110±.0030	.6865±.0106	.1080±.0798	.1482±.1111
	cos-LDL	.0123±.0039	.0201±.0036	.9407±.0117	.9874±.0041	.3134±.0544	.7576±.2015	.1443±.1328	.1828±.1668
	LALOT	.0084±.0003	.0165±.0003	.9526±.0008	.9917±.0003	.2608±.0043	.8544±.0150	.0322±.0050	.0451±.0069
	LDLSF	.0058±.0001	.0139±.0002	.9613±.0004	.9943±.0001	.2164±.0021	.6874±.0111	.1005±.0132	.1160±.0111
	LDL-SCL	.0055±.0001	.0135±.0002	.9620±.0005	.9945±.0001	.2113±.0032	.6868±.0084	.1223±.0056	.1388±.0084
	MSLP	.0056±.0001	.0136±.0002	.9618±.0005	.9944±.0001	.2126±.0031	.6870±.0118	.1700±.0118	.1839±.0118
	PT-Bayes	.3046±.0090	.1054±.0023	.7624±.0037	.8391±.0032	1.2237±.0182	.6896±.0062	.0271±.0094	.0299±.0062
	AA-kNN	.0064±.0001	.0148±.0001	.9581±.0005	.9935±.0001	.2315±.0030	.6928±.0155	.1222±.0131	.1411±.0155
	CPNN	.0058±.0002	.0137±.0002	.9610±.0007	.9943±.0001	.2165±.0045	.6998±.0558	.0362±.0100	.0463±.0558
cdc	LDL-LRR	**.0067±.0002**	**.0160±.0003**	**.9582±.0006**	**.9935±.0002**	**.2126±.0044**	**.4277±.0115**	**.1956±.0105**	**.1930±.0115**
	BFGS-LLD	.0071±.0002	.0163±.0002	.9570±.0007	.9932±.0002	.2175±.0031	.6535±.0101	.1217±.0529	.1671±.0723
	cos-LDL	.0102±.0042	.0198±.0046	.9480±.0124	.9900±.0045	.2577±.0532	.7829±.1772	.0653±.0733	.0880±.1006
	LALOT	.0088±.0003	.0187±.0004	.9524±.0006	.9915±.0002	.2422±.0037	.7224±.0087	-.0174±.0090	-.0223±.0119
	LDLSF	.0070±.0002	.0162±.0002	.9573±.0006	.9933±.0002	.2163±.0025	.4287±.0163	.1789±.0199	.1878±.0163
	LDL-SCL	.0068±.0002	.0161±.0003	**.9582±.0005**	.9934±.0002	.2131±.0038	.4300±.0095	.1002±.0122	.0964±.0095
	MSLP	.0070±.0002	.0162±.0003	.9573±.0006	.9932±.0002	.2165±.0037	.4365±.0129	.1566±.0113	.1670±.0129
	PT-Bayes	.2938±.0063	.1120±.0011	.7671±.0023	.8460±.0016	1.0969±.0106	.4286±.0066	.0167±.0056	.0189±.0066
	AA-kNN	.0080±.0004	.0176±.0003	.9532±.0011	.9921±.0004	.2357±.0066	.4390±.0211	.1367±.0135	.1499±.0211
	CPNN	.0074±.0001	.0167±.0002	.9567±.0006	.9929±.0001	.2203±.0010	.4411±.0355	.0723±.0256	.0843±.0355

↑ (↓) *indicates the higher (lower), the better. The best results on each row are highlighted.*

significance level on all results. These comparisons are summarized in Table 5, which presents the win/tie/loss counts of the LDL-LRR algorithm on each metric for the row over the algorithm for the column. In summary, our proposal is clearly always significantly better than or comparable to all other algorithms, except AA-kNN and LDL-SCL obtain a better result on metric *Clark*, respectively. In detail, compared with CPNN, AA-kNN, and PT-Bayes, our proposed LDL-LRR has a significant advantage on all the evaluation metrics. Compared with MSLP, LDL-SCL, and LDLSF,

TABLE 4
Comparison Results on Six Datasets are Shown as "mean ± std"

data	algorithm	*K-L*↓	*Chebyshev*↓	*Intersection*↑	*Cosine*↑	*Clark*↓	*Canberra*↓	τ_K↑	ρ_S↑
elu	LDL-LRR	**.0061 ± .0002**	**.0162 ±. 0002**	**.9588 ±. 0005**	**.9939 ±. 0002**	**.2000 ±. 0026**	**.5846 ±. 0085**	**.1988 ±. 0088**	**.2030 ±.0110**
	BFGS-LLD	.0062 ±. 0003	.0163 ±. 0004	.9587 ±. 0007	.9940 ±. 0002	.2012 ±. 0045	.5847 ±. 0113	.1651 ±. 0066	.1936 ±. 0085
	cos-LDL	.0113 ±. 0040	.0224 ±. 0049	.9436 ±. 0115	.9886 ±. 0043	.2633 ±. 0485	.7861 ±. 1526	.0724 ±. 0683	.1011 ±. 0917
	LALOT	.0067 ±. 0002	.0169 ±. 0002	.9566 ±. 0006	.9935 ±. 0002	.2088 ±. 0029	.6148 ±. 0093	.0126 ±. 0102	.0184 ±. 0150
	LDLSF	.0063 ±. 0002	.0164 ±. 0003	.9583 ±. 0006	.9938 ±. 0002	.2003 ±. 0026	.5966 ±. 0083	.1790 ±. 0108	.1902 ±. 0127
	LDL-SCL	.0062 ±. 0001	.0164 ±. 0003	.9586 ±. 0003	.9938 ±. 0002	.2001 ±. 0038	.5848 ±. 0075	.1019 ±. 0101	.0980 ±. 0065
	MSLP	.0063 ± .0001	.0164 ± .0003	.9581 ± .0005	.9938 ± .0001	.2024v ± .0025	.5931 ± .0079	.1589 ± .0131	.1689 ± .0132
	PT-Bayes	.2038 ± .0053	.1130 ± .0010	.7876 ± .0026	.8666 ± .0036	1.0009 ± .0100	.6606 ± .0036	.0176 ± .0086	.0198 ± .0055
	AA-kNN	.0070 ± .0002	.0166 ± .0003	.9502 ± .0010	.9933 ± .0004	.2333 ± .0060	.6280 ± .0211	.1378 ± .0136	.1503 ± .0201
	CPNN	.0077 ± .0001	.0169 ± .0002	.9468 ± .0005	.9920 ± .0001	.2353 ± .0010	.6411 ± .0305	.0733 ± .0156	.0855 ± .0101
diau	LDL-LRR	**.0127 ± .0004**	**.0366 ± .0004**	**.9410 ± .0010**	**.9881 ± .0004**	**.1977 ± .0028**	**.4277 ± .0067**	**.5002 ± .0087**	**.4081 ± .0057**
	BFGS-LLD	.0133 ± .0008	.0371 ± .0014	.9399 ± .0020	.9878 ± .0007	.2015 ± .0064	.4330 ± .0137	.3037 ± .1020	.3770 ± .1268
	cos-LDL	.0201 ± .0040	.0471 ± .0058	.9231 ± .0095	.9808 ± .0041	.2504 ± .0276	.5459 ± .0631	-.0009 ± .0667	-.0031 ± .0926
	LALOT	.0161 ± .0005	.0412 ± .0008	.9319 ± .0010	.9850 ± .0004	.2247 ± .0040	.4885 ± .0073	-.0260 ± .0166	-.0310 ± .0195
	LDLSF	.0139 ± .0003	.0380 ± .0007	.9387 ± .0009	.9872 ± .0003	.2058 ± .0029	.4300 ± .0101	.3512 ± .0132	.3897v ± .0120
	LDL-SCL	.0131 ± .0005	.0370 ± .0006	.9405 ± .0010	.9879 ± .0004	.1998 ± .0035	.4287 ± .0122	.2691 ± .0129	.2965 ± .0123
	MSLP	.0130 ± .0004	.0369 ± .0006	.9406 ± .0010	.9879 ± .0004	.1991v ± .0035	.4286 ± .0132	.4521 ± .0189	.4005 ± .0162
	PT-Bayes	.3024 ± .0109	.1683 ± .0032	.7611 ± .0044	.8553 ± .0031	.7799 ± .0151	.4290 ± .0200	.1023 ± .0189	.1292 ± .0200
	AA-kNN	.0153 ± .0008	.0399 ± .0010	.9360 ± .0017	.9859±.0007	.2143 ± .0057	.4320 ± .0164	.3099 ± .0211	.3477 ± .0274
	CPNN	.0146 ± .0005	.0400 ± .0007	.9361 ± .0010	.9865 ± .0004	.2140 ± .0042	.4365 ± .0104	.2233 ± .0198	.2254 ± .0114
heat	LDL-LRR	**.0125 ± .0005**	**.0418 ± .0006**	**.9406 ± .0007**	**.9881 ± .0004**	**.1816 ± .0032**	**.3618 ± .0066**	**.1988 ± .0086**	**.1812 ± .0067**
	BFGS-LLD	.0130 ± .0006	.0430 ± .001	.9394 ± .0012	.9876 ± .0006	.1849 ± .0039	.3688 ± .0074	.0989 ± .6050	.1211 ± .0752
	cos-LDL	.0136 ± .0014	.0439 ± .0025	.9376 ± .0042	.9869 ± .0015	.1893 ± .0104	.3792 ± .0234	.1126 ± .1057	.1272 ± .1118
	LALOT	.0148 ± .0006	.0456 ± .0010	.9362 ± .0012	.9859 ± .0005	.1940 ± .0036	.3881 ± .0069	-.0062 ± .0144	-.0027 ± .0179
	LDLSF	.0127 ± .0009	.0603 ± .0015	.9192 ± .0016	.9784 ± .0009	.2427 ± .0045	.3666 ± .0064	.0356 ± .0082	.0258 ± .0064
	LDL-SCL	**.0125 ± .0004**	.0420 ± .0007	**.9406 ± .0007**	**.9881 ± .0004**	.1818 ± .0028	.3687 ± .0126	.1856 ± .0118	.1810 ± .0126
	MSLP	.0133 ± .0004	.0431 ± .0007	.9385 ± .0006	.9873 ± .0003	.1873 ± .0030	.3619 ± .0115	.1392 ± .0092	.1406 ± .0115
	PT-Bayes	.2746 ± .0119	.1756 ± .0034	.7690 ± .0039	.8640 ± .0032	.6890 ± .0096	.3662 ± .0108	.0726 ± .0120	.0693 ± .0117
	AA-kNN	.0142 ± .0008	.0446 ± .0013	.9363 ± .0017	.9864 ± .0007	.1932 ± .0054	.3678 ± .0085	.1395 ± .0136	.1592 ± .0385
	CPNN	.0213 ± .0118	.0527 ± .0138	.9230 ± .0229	.9799 ± .0111	.2305 ± .0622	.3689 ± .0593	.0988 ± .0182	.0880 ± .0593
spo5	LDL-LRR	**.0297 ± .0012**	**.0919 ± .0019**	**.9080 ± .0010**	**.9738 ± .0009**	**.1860 ± .0039**	**.5115 ± .0036**	**.1361 ± .0112**	**.1385 ± .0236**
	BFGS-LLD	.0299 ± .0009	.0924 ± .0014	.9077 ± .0014	.9732 ± .0007	.1875 ± .0033	.5821 ± .0045	.0670 ± .0412	.0654 ± .0433
	cos-LDL	.0307 ± .0030	.0933 ± .0052	.9067 ± .0052	.9729 ± .0029	.1876 ± .0088	.5886 ± .0148	.0273 ± .0223	.0029 ± .0268
	LALOT	.0313 ± .0015	.0949 ± .0024	.9051 ± .0024	.9723 ± .0012	.1903 ± .0049	.5938 ± .0076	.0608 ± .0154	.0731 ± .0202
	LDLSF	.0491 ± .0015	.1259 ± .0019	.8741 ± .0019	.9544 ± .0022	.2435 ± .0040	.5144 ± .0087	.0100 ± .0201	.0100 ± .0297
	LDL-SCL	.0299 ± .0011	.0923 ± .0019	.9076 ± .0009	.9736 ± .0009	.1865 ± .0039	.5144 ± .0193	.1360 ± .0193	.1360 ± .0193
	MSLP	.0311 ± .0015	.0935 ± .0023	.9064 ± .0013	.9725 ± .0012	.1891 ± .0050	.5203 ± .0031	.1152 ± .0190	.1379 ± .0331
	PT-Bayes	.2310 ± .0118	.2139 ± .0060	.7861 ± .0055	.8883 ± .0042	.4456 ± .0127	.5210 ± .0129	.0383 ± .0122	.0459 ± .0229
	AA-kNN	.0330 ± .0028	.0954 ± .0042	.9046 ± .0028	.9708 ± .0024	.1920 ± .0081	.5299 ± .0042	.1010 ± .0100	.0954 ± .0042
	CPNN	.0373 ± .0073	.1059 ± .0128	.8941 ± .0104	.9666 ± .0069	.2090 ± .0196	.5310 ± .0059	.0183 ± .0321	.0183v ± .0359
cold	LDL-LRR	**.0120 ± .0005**	**.0506 ± .0010**	**.9415 ± .0012**	**.9887 ± .0005**	**.1381 ± .0032**	**.2393 ± .0048**	**.2389 ± .0165**	**.2469 ± .0188**
	BFGS-LLD	.0125 ± .0007	.0519 ± .0013	.9401 ± .0015	.9883 ± .0005	.1411 ± .0039	.2429 ± .0064	.1914 ± .0657	.2256 ± .0751
	cos-LDL	.0146 ± .0032	.0564 ± .0065	.9347 ± .0073	.9862 ± .0030	.1530 ± .0165	.2641 ± .0285	.0963 ± .0282	.1382 ± .0406
	LALOT	.0153v ± .0008	.0581 ± .0013	.9330 ± .0017	.9856 ± .0008	.1571 ± .0041	.2712 ± .0073	-.1440 ± .0180	-.1749 ± .0187
	LDLSF	.0130 ± .0005	.0530 ± .0004	.9389 ± .0009	.9878 ± .0004	.1443 ± .0017	.2399 ± .0069	.2301 ± .0109	.2128 ± .0129
	LDL-SCL	.0141 ± .0006	.0552 ± .0010	.9361 ± .0012	.9867 ± .0005	.1500 ± .0031	.2488 ± .0088	-.0501 ± .0221	-.0266 ± .0138
	MSLP	.0131 ± .0005	.0529 ± .0012	.9386 ± .0016	.9876 ± .0005	.1444 ± .0037	.2397 ± .0070	.2039 ± .0170	.2238 ± .0270
	PT-Bayes	.2361 ± .0213	.1900 ± .0095	.7886 ± .0098	.8867 ± .0069	.5141 ± .0233	.2431 ± .0084	.0512 ± .0172	.0420 ± .0214
	AA-kNN	.0141 ± .0013	.0557 ± .0022	.9356 ± .0025	.9866 ± .0011	.1512 ± .0059	.2486 ± .0053	.2100 ± .0103	.2190 ± .0253
	CPNN	.0127 ± .0007	.0523 ± .0016	.9394 ± .0017	.9880 ± .0006	.1425 ± .0044	.2487 ± .0070	.2310 ± .0100	.2292 ± .0170
dtt	LDL-LRR	**.0061 ± .0002**	**.0359 ± .0007**	**.9584 ± .0008**	**.9941 ± .0002**	**.0978 ± .0017**	**.1682 ± .0030**	**.2082 ± .0280**	**.2038 ± .0330**
	BFGS-LLD	.0064 ± .0004	.0364 ± .0009	.9578 ± .0010	.9939 ± .0003	.0993 ± .0027	.1708 ± .0042	.1413 ± .0347	.1633 ± .0411
	cos-LDL	.0092 ± .0029	.0437 ± .0071	.9492 ± .0086	.9913 ± .0028	.1185 ± .0192	.2050 ± .0342	.0373 ± .0354	.0537 ± .0517
	LALOT	.0080 ± .0003	.0420 ± .0006	.9521 ± .0007	.9924 ± .0002	.1134 ± .0014	.1941 ± .0026	-.0367 ± .0284	-.0434 ± .0350
	LDLSF	.0063 ± .0003	.0360 ± .0007	.9583 ± .0008	**.9941 ± .0002**	.0983 ± .0019	.1691 ± .0054	.2056 ± .0176	.1992 ± .0254
	LDL-SCL	.0062 ± .0002	.0360 ± .0006	.9582 ± .0007	.9940 ± .0001	.0983 ± .0195	.1717 ± .0092	.1860 ± .0221	.2033 ± .0332
	MSLP	.0062 ± .0002	**.0359 ± .0007**	.9583 ± .0007	**.9941 ± .0002**	.0979 ± .0016	.1751 ± .0084	.1572 ± .0162	.1941 ± .0184
	PT-Bayes	.2309 ± .0139	.1836 ± .0060	.7955 ± .0063	.8920 ± .0042	.4980 ± .0139	.1688 ± .0066	.0501 ± .0100	.0691 ± .0136
	AA-kNN	.0074 ± .0007	.0395 ± .0012	.9544 ± .0012	.9929 ± .0006	.1076 ± .0032	.1699 ± .0078	.1027 ± .0211	.1321 ± .0378
	CPNN	.0068 ± .0005	.0378 ± .0016	.9564 ± .0016	.9936 ± .0004	.1028 ± .0041	.1751 ± .0061	.0411 ± .0212	.0427±.0621

↑ (↓) indicates the higher (lower), the better. The best results on each row are highlighted.

TABLE 5
Summary of the Comparison (win/tie/loss) of LDL_LRR on Each Metric for the Row Over Other Methods for the Column, Under the Pairwise Two-Tailed t-Test With 0.05 Significance Level

Measures	CPNN	AA-kNN	PT-Bayes	MSLP	LDL-SCL	LDLSF	LALOT	cos-LDL	BFGS-LLD
K-L	12/1/0	13/0/0	13/0/0	10/3/0	6/7/0	9/4/0	12/1/0	12/1/0	5/8/0
Chebyshev	12/1/0	13/0/0	13/0/0	9/4/0	5/8/0	8/4/1	12/1/0	12/1/0	7/6/0
Intersection	12/1/0	11/2/0	13/0/0	10/3/0	6/7/0	9/4/0	13/0/0	12/1/0	8/5/0
Cosine	12/1/0	13/0/0	13/0/0	9/4/0	5/8/0	9/4/0	12/1/0	12/1/0	9/4/0
Clark	13/0/0	11/1/1	13/0/0	8/5/0	4/8/1	10/3/0	11/2/0	12/1/0	8/5/0
Canberra	10/3/0	9/4/0	8/5/0	7/6/0	5/8/0	6/7/0	11/2/0	10/3/0	10/3/0
ρ_S	13/0/0	13/0/0	13/0/0	10/3/0	9/4/0	11/2/0	13/0/0	13/0/0	11/2/0
τ_K	12/1/0	13/0/0	13/0/0	13/0/0	11/2/0	11/2/0	13/0/0	13/0/0	11/2/0

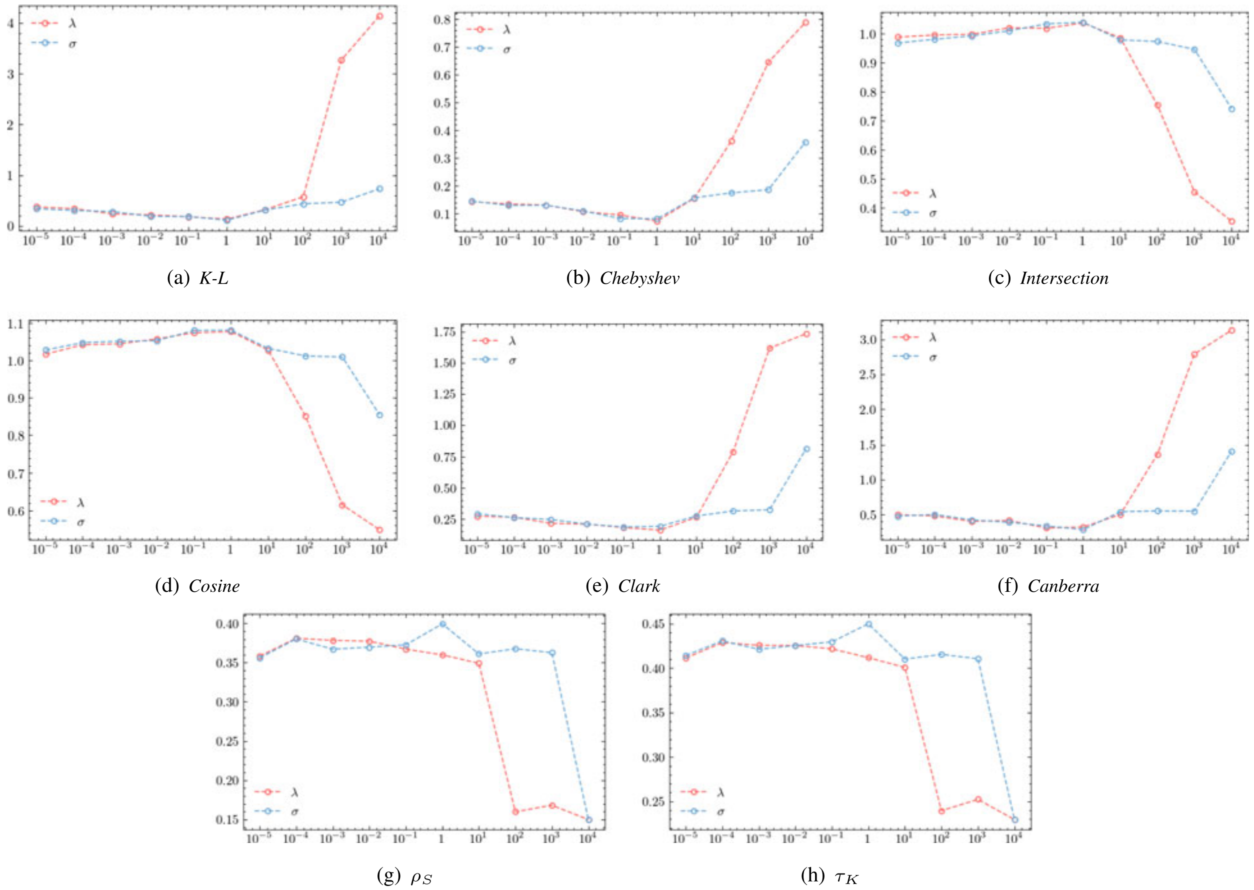
LDL-LRR has a slight advantage on the six traditional evaluation metrics but a significant improvement on the two ranking-based metrics.

Ablation Studies. In this part, we perform ablation experiment on all datasets in order to better understand the importance of our proposed ranking loss function. The "Baseline"

TABLE 6
Ablation Results on 13 Datasets are Shown as "mean±std"

data	task	*K-L*↓	*Chebyshev*↓	*Intersection*↑	*Cosine*↑	*Clark*↓	*Canberra*↓	ρ_S↑	τ_K↑
sj	Baseline	0.0568	0.0968	0.8630	0.9468	0.3839	0.8821	0.4668	0.4879
	Baseline + LR	0.0376	0.0817	0.8910	0.9647	0.3133	0.6421	0.5568	0.6678
mov	Baseline	0.1156	0.1221	0.8268	0.9338	0.5356	1.1056	0.6163	0.6218
	Baseline + LR	0.0982	0.1151	0.8355	0.9353	0.5235	0.9996	0.7070	0.7053
ns	Baseline	0.9201	0.3130	0.4886	0.6789	2.3560	6.8116	0.3270	0.5520
	Baseline + LR	0.7314	0.2997	0.5670	0.7532	2.4268	5.7143	0.5301	0.6001
gene	Baseline	0.2930	0.0621	0.7459	0.7719	2.3663	14.8210	0.1003	0.1129
	Baseline + LR	0.2365	0.0532	0.7844	0.8346	2.1114	13.5681	0.1618	0.1808
emo	Baseline	0.8567	0.3319	0.5518	0.6629	1.7135	0.7016	0.1789	0.3018
	Baseline + LR	0.5966	0.3109	0.5802	0.7086	1.6599	0.6819	0.3544	0.3523
alpha	Baseline	0.0058	0.0139	0.9601	0.9934	0.2175	0.6999	0.1231	0.1997
	Baseline + LR	0.0054	0.0134	0.9625	0.9946	0.2093	0.6791	0.2144	0.2058
cdc	Baseline	0.0075	0.0169	0.9558	0.9920	0.2205	0.4417	0.1009	0.1802
	Baseline + LR	0.0067	0.0160	0.9582	0.9935	0.2126	0.4277	0.1930	0.1956
elu	Baseline	0.0078	0.0170	0.9466	0.9918	0.2208	0.2225	0.1028	0.1257
	Baseline + LR	0.0061	0.0162	0.9588	0.9939	0.2000	0.5846	0.2030	0.1988
diau	Baseline	0.0149	0.0414	0.9339	0.9856	0.2145	0.4372	0.3155	0.4190
	Baseline + LR	0.0127	0.0366	0.9410	0.9881	0.1977	0.4277	0.4081	0.5002
heat	Baseline	0.0213	0.0528	0.9228	0.9769	0.2305	0.3693	0.1681	0.1039
	Baseline + LR	0.0125	0.0418	0.9406	0.9881	0.1816	0.3618	0.1812	0.1988
spo5	Baseline	0.0376	0.1060	0.8940	0.9564	0.2093	0.5319	0.1237	0.0974
	Baseline + LR	0.0297	0.0919	0.9080	0.9738	0.1860	0.5115	0.1385	0.1361
cold	Baseline	0.0450	0.0686	0.9365	0.9786	0.1519	0.2418	0.2208	0.2110
	Baseline + LR	0.0376	0.0506	0.9415	0.9887	0.1381	0.2393	0.2469	0.2389
dtt	Baseline	0.0073	0.0394	0.9540	0.9923	0.1029	0.1755	0.1863	0.1027
	Baseline + LR	0.0061	0.0359	0.9584	0.9941	0.0978	0.1682	0.2038	0.2082

↑ (↓) indicates the higher (lower), the better. "Baseline" is trained without the label ranking loss term (i.e., setting the λ coefficient of the first term of Eq. 11 to 0). "Baseline + LR" is trained with the label ranking loss term (i.e., setting the λ coefficient of the first term of Eq. 11 to an appropriate number).



(a) *K-L* (b) *Chebyshev* (c) *Intersection*

(d) *Cosine* (e) *Clark* (f) *Canberra*

(g) ρ_S (h) τ_K

Fig. 4. Influence of λ and σ with 8 metrics on dataset Yeast_cold

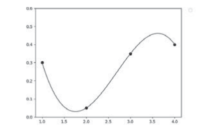
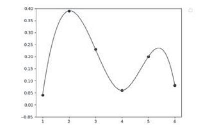
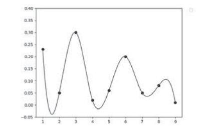
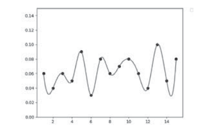
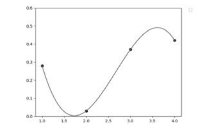
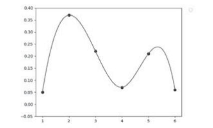
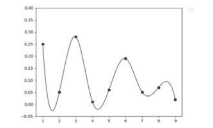
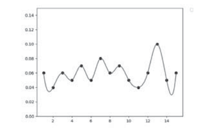
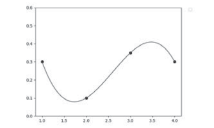
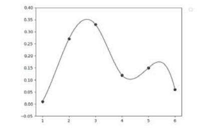
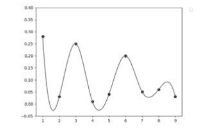
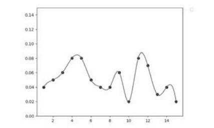
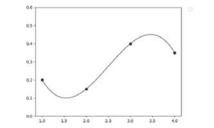
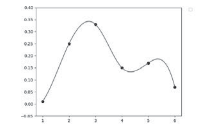
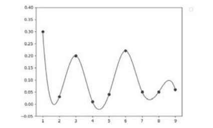
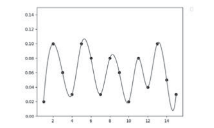
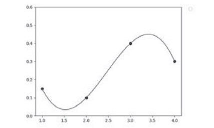
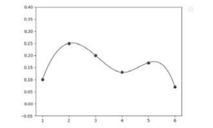
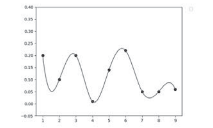
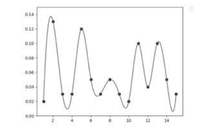
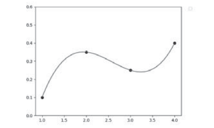
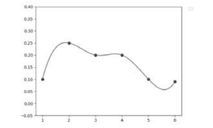
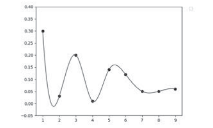
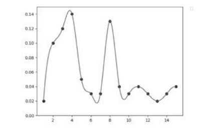
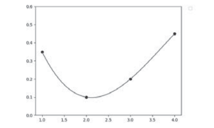
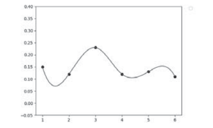
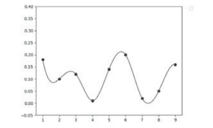
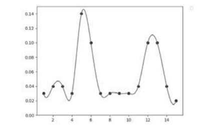
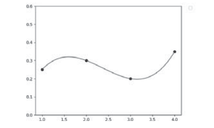
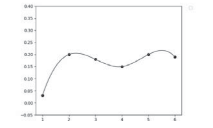
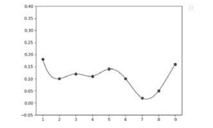
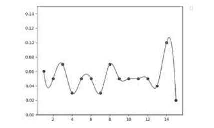
method is trained without the label ranking loss term, i.e., setting the coefficient $\lambda = 0$ in Eq. (11). The "Baseline + LR" method (our proposed LDL-LRR) considers the label ranking loss term.

As shown in Table 6, the task "Baseline + LR" has a huge improvement on most of the evaluation metrics compared to "Baseline". This result confirms the validity and reasonableness of the label ranking relation for LDL.

Influence of Parameter. In order to examine the robustness of the algorithm, we also analyze the influence of parameters in the experiment, i.e., λ and σ in Eq. (11). For the analysis of λ, we fix σ to 100 and run LDL-LRR with λ which is set to $10^{-5}, 10^{-4}, \ldots, 10^{4}$. For the analysis of σ, we fix λ to 1 and run LDL-LRR with σ, similar to λ, which is set to $10^{-5}, 10^{-4}, \ldots, 10^{4}$. Due to the page limit, we only present the experimental results on dataset Yeast_cold with 8 evaluation metrics. The results are shown in Fig. 4. Notice that, for metrics *Canberra*, *Chebyshev*, *Clark* and *K-L*, the smaller the value, the better the performance; but for metrics *Intersection*, *Cosine*, ρ_S and τ_K, the larger the value, the better the performance. As for parameter λ shown in Fig. 4, we can see that when the value of λ is 0.01, 0.1 or 1, the performance is best, and when λ takes a smaller value (such as 0.0001, 0.001) or a larger value (such as 1000, 10000), the performance gets worse on all eight evaluation metrics. This is because when λ is very small, the ranking loss term we proposed plays a small role, which also indicates the effectiveness of the loss function we proposed. And when λ is too large, the objective function is not dominated by the third term in Eq. (11). Similarly, the results of σ has the same trend to λ since it also controls the importance of ranking loss term.

Case Study. In addition, to more intuitively illustrate the role of the label ranking relation in the fitting process, due to the limited space of the paper, we select four representative samples from different datasets, and Table 7 shows its prediction results by different algorithms. The first row shows the ground-truth label distributions of four typical test instances with 4, 6, 9 and 15 labels. Each of the following rows shows the corresponding predictions of one LDL algorithm. We can clearly see that the curve fitting LDL-LRR and LDL-SCL is very similar to the ground truth distribution curve, LDLSF, AA-KNN and MSLP are slightly similar, and CPNN and PT-Bayes are poorly fitted. Moreover, the algorithms LDL-LRR and LDL-SCL are similar to each other on *K-L* and *Cosine*, while the LDL-LRR with higher *Spearman's rank* obviously fits the ground-truth curve better.

TABLE 7
Some Typical Examples Illustrate the Role of the Label Ranking Relation in Fitting the Ground-Truth Label Distribution

Ground-truth	 4 labels	 6 labels	 9 labels	 15 labels
LDL-LRR	 {0.002,0.998,1}	 {0.001,0.997,0.970}	 {0.001,0.998,1}	 {0.002,0.996,0.962}
LDL-SCL	 {0.002,0.998,0.964}	 {0.001,0.996,0.956}	 {0.001,0.997,0.956}	 {0.003,0.995,0.893}
LDLSF	 {0.004,0.996,0.954}	 {0.001,0.995,0.956}	 {0.001,0.997,0.906}	 {0.004,0.993,0.723}
AA-kNN	 {0.008,0.997,0.920}	 {0.005,0.994,0.925}	 {0.002,0.996,0.884}	 {0.009,0.990,0.589}
MSLP	 {0.041,0.991,0.664}	 {0.021,0.989,0.650}	 {0.003,0.991,0.739}	 {0.015,0.982,-0.034}
CPNN	 {0.090,0.957,0.784}	 {0.056,0.937,0.739}	 {0.016,0.723,0.009}	 {0.072,0.959,-0.147}
PT-Bayes	 {0.103,0.950,-0.015}	 {0.093,0.957,-0.005}	 {0.120,0.791,0.031}	 {0.122,0.947,-0.158}

The horizontal axis represents the emotion labels, and the vertical axis represents the corresponding emotion description value of each label. The solid origin on the curve is the value of label distribution. The numbers below each predicted label distribution correspond to three evaluation metrics: K-L, Cosine, *and* Spearman's rank.

5 Conclusion

As a generalization of multi-label learning, label distribution learning can deal with label ambiguity problems by considering more label information. In this paper, to improve the effectiveness of LDL, we propose a novel LDL method by exploiting label ranking relation. We introduce a ranking loss function for LDL to maintain the label ranking relation, and propose a novel LDL algorithm LDL-LRR. In addition, we also argue that the existing distance-based or similarity-based evaluation metrics are not sufficient to fully measure the validity of LDL algorithms, for which we introduce two ranking-based evaluation metrics for the first time to evaluate the label ranking relation in LDL. The experimental results on 13 real-world datasets validate the effectiveness of our proposal.

In future work, we will investigate in more depth the role of label ranking in LDL. We will study how to apply more different kinds of ranking loss functions on LDL to improve the generalization ability of LDL models. In addition, we will also explore the impact of the ranking loss functions on LDL when the label distributions have different uncertainties.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1706900, in part by the National Natural Science Foundation of China under Grants 61906090 and U20B2064, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20191287, and in part by the Fundamental Research Funds for Central Universities under Grant 30920021131.

References

[1] G. Tsoumakas, I. Katakis, and D. Taniar, "Multi-label classification: An overview," *Int. J. Data Warehousing Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[2] X. Geng, "Label distribution learning," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1734–1748, Jul. 2016.

[3] X. Jia, W. Li, J. Liu, and Y. Zhang, "Label distribution learning by exploiting label correlations," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3310–3317.

[4] Y. Ren and X. Geng, "Sense beauty by label distribution learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 2648–2654.

[5] B. Gao, H. Zhou, J. Wu, and X. Geng, "Age estimation using expectation of label distribution learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 712–718.

[6] C. Peng, A. Tao, and X. Geng, "Label embedding based on multiscale locality preservation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 2623–2629.

[7] K. Wang and X. Geng, "Binary coding based label distribution learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 2783–2789.

[8] P. Li, C. J. C. Burges, and Q. Wu, "Mcrank: Learning to rank using multiple classification and gradient boosting," in *Proc. Neural Inf. Process. Syst.*, 2007, pp. 897–904.

[9] C. Marsala, M. Detyniecki, N. Usunier, and M. Amini, "Uhigh-level feature detection with forests of fuzzy decision trees combined with the rankboost algorithm," in *Proc. TRECVID Workshop*, 2007, p. 10.

[10] H. Valizadegan, R. Jin, R. Zhang, and J. Mao, "Learning to rank by optimizing NDCG measure," in *Proc. Neural Inf. Process. Syst.*, 2009, pp. 1883–1891.

[11] X. Geng, K. Smith-Miles, and Z.-H. Zhou, "Facial age estimation by learning from label distributions," in *Proc. AAAI Conf. Artif. Intell.*, 2010, pp. 451–456.

[12] X. Chao, G. Xin, and X. Hui, "Logistic boosting regression for label distribution learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4489–4497.

[13] X. Zheng, X. Jia, and W. Li, "Label distribution learning by exploiting sample correlations locally," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4556–4563.

[14] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing," *Comput. Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[15] X. Jia, X. Zheng, W. Li, C. Zhang, and Z. Li, "Facial emotion distribution learning by exploiting low-rank label correlations locally," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9841–9850.

[16] X. Geng, Q. Wang, and Y. Xia, "Facial age estimation by adaptive label distribution learning," in *Proc. Int. Conf. Pattern Recognit.*, 2014, pp. 4465–4470.

[17] D. Zhou, X. Zhang, Y. Zhou, Q. Zhao, and X. Geng, "Emotion distribution learning from texts," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 638–647.

[18] X. Geng and Y. Xia, "Head pose estimation based on multivariate label distribution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1837–1842.

[19] Y. Zhou, H. Xue, and X. Geng, "Emotion distribution recognition from facial expressions," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 1247–1250.

[20] T. Ren, X. Jia, W. Li, L. Chen, and Z. Li, "Label distribution learning with label-specific features," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 3318–3324.

[21] J. Wang and X. Geng, "Classification with label distribution learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 3712–3718.

[22] D. Cossock and T. Zhang, "Subset ranking using regression," in *Proc. Int. Conf. Comput. Learn. Theory*, 2006, pp. 605–619.

[23] H. Drucker, B. Shahrary, and D. C. Gibbon, "Support vector machines: Relevance feedback and information retrieval," *Inf. Process. Manage.*, vol. 38, no. 3, pp. 305–323, 2002.

[24] P. Li, Q. Wu, and C. Burges, "McRank: Learning to rank using multiple classification and gradient boosting," in *Proc. Conf. Neural Inf. Process. Syst.*, 2008, pp. 897–904.

[25] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2002, pp. 133–142.

[26] C. Burges *et al.*, "Learning to rank using gradient descent," in *Proc. Int. Conf. Mach. Learn.*, 2005, pp. 129–136.

[27] Z. Zheng, H. Zha, T. Zhang, O. Chapelle, K. Chen, and G. Sun, "A general boosting method and its application to learning ranking functions for web search," in *Proc. Neural Inf. Process. Syst.*, 2008, pp. 1697–1704.

[28] C. Burges, K. Svore, P. N. Bennett, A. Pastusiak, and Q. Wu, "Learning to rank using an ensemble of Lambda-gradient models," in *Proc. Int. Conf. Yahoo! Learn. Rank Challenge*, 2010, pp. 25–35.

[29] C. Damke and E. Hullermeier, "Ranking structured objects with graph neural networks," 2021, *arXiv:2104.08869*.

[30] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: From pairwise approach to listwise approach," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 129–136.

[31] M. Köppel, A. Segner, M. Wagener, L. Pensel, A. Karwath, and S. Kramer, "Pairwise learning to rank by neural networks revisited: Reconstruction, theoretical analysis and practical performance," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2020, pp. 237–252.

[32] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation methods for the web," in *Proc. Int. Conf. World Wide Web*, 2001, pp. 613–622.

[33] X. Chang, Y.-L. Yu, Y. Yang, and E. P. Xing, "They are not equally reliable: Semantic event search using differentiated concept classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1884–1893.

[34] X. Chang, Y. Yang, A. G. Hauptmann, E. P. Xing, and Y.-L. Yu, "Semantic concept discovery for large-scale zero-shot event detection," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 2234–2240.

[35] S. H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *Int. J. Math. Models Methods Appl. Sci.*, vol. 1, no. 4, pp. 300–307, 2007.

[36] C. J. C. Burges *et al.*, "Learning to rank using gradient descent," in *Proc. Int. Conf. Mach. Learn.*, 2005, pp. 89–96.

[37] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," in *Proc. Int. Conf. Learn. Representations*, 2018, p. 37.

[38] M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with Gabor wavelets," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, 1998, pp. 200–205.

[39] K. C. Peng, T. Chen, A. Sadovnik, and A. Gallagher, "A mixed bag of emotions: Model, predict, and transfer emotion distributions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 860–868.

[40] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," in *Proc. Nat. Acad. Sci. United States Amer.*, 1998, pp. 14863–14868.

[41] X. Geng and L. Luo, "Multilabel ranking with inconsistent rankers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3742–3747.

[42] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, 2004.

[43] C. Spearman, "The proof and measurement of association between two things," *Amer. J. Psychol.*, vol. 100, no. 3/4, pp. 441–471, 1987.

[44] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[45] H. ru Zhang, Y. Ting Huang, Y. Yuan Xu, and F. Min, "Cos-LDL: Label distribution learning by Cosine-based distance-mapping correlation," *IEEE Access*, vol. 8, pp. 63961–63970, Mar. 2020.

[46] P. Zhao and Z.-H. Zhou, "Label distribution learning by optimal transport," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4506–4513.

[47] X. Geng, C. Yin, and Z.-H. Zhou, "Facial age estimation by learning from label distributions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2401–2412, Oct. 2013.

Xiuyi Jia (Member, IEEE) received the PhD degree in computer science from Nanjing University, Nanjing, China, in 2011. He is currently an associate professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include machine learning, data mining, and computer vision.

vXiaoxia Shen received the master's degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology. Her research interests include machine learning and data mining.

Weiwei Li received the PhD degree in computer science from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2016. She is currently an associate professor with the Nanjing University of Aeronautics and Astronautics. Her research interests include machine learning, software data mining, and knowledge engineering.

Yunan Lu is currently working toward the PhD degree with the Nanjing University of Science and Technology. His research interests include machine learning and data mining.

Jihua Zhu (Member,IEEE) received the BE degree in automation from Central South University, China, in 2004, and the PhD degree in pattern recognition and intelligence systems from Xi'an Jiaotong University, China, in 2011. He is currently an associate professor with the School of Software Engineering, Xi'an Jiaotong University. His research interests include computer vision and machine learning.