Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 1090565, 11 pages
https://doi.org/10.1155/2018/1090565

Research Article

Label Distribution Learning by Regularized Sample Self-Representation

Wenyuan Yang, Chan Li, and Hong Zhao

Lab of Granular Computing, Minnan Normal University, Zhangzhou, Fujian 363000, China

Correspondence should be addressed to Wenyuan Yang; yangwy@xmu.edu.cn

Received 19 October 2017; Revised 1 January 2018; Accepted 12 March 2018; Published 23 April 2018

Academic Editor: Wanquan Liu

Multilabel learning that focuses on an instance of the corresponding related or unrelated label can solve many ambiguity problems. Label distribution learning (LDL) reflects the importance of the related label to an instance and offers a more general learning framework than multilabel learning. However, the current LDL algorithms ignore the linear relationship between the distribution of labels and the feature. In this paper, we propose a regularized sample self-representation (RSSR) approach for LDL. First, the label distribution problem is formalized by sample self-representation, whereby each label distribution can be represented as a linear combination of its relevant features. Second, the LDL problem is solved by L_2-norm least-squares and $L_{2,1}$-norm least-squares methods to reduce the effects of outliers and overfitting. The corresponding algorithms are named RSSR-LDL2 and RSSR-LDL21. Third, the proposed algorithms are compared with four state-of-the-art LDL algorithms using 12 public datasets and five evaluation metrics. The results demonstrate that the proposed algorithms can effectively identify the predictive label distribution and exhibit good performance in terms of distance and similarity evaluations.

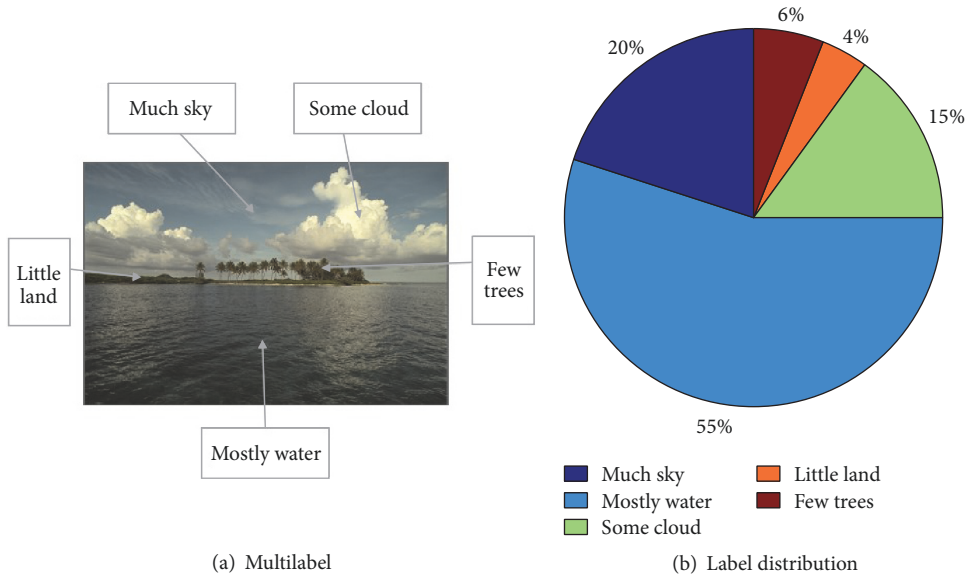
1. Introduction

Multilabel learning allows more than one label to be associated with each instance [1]. In many practical applications, such as text categorization, ticket sales, and torch relays [2], objects have more than one semantic label, often expressed as the objects ambiguity. As an effective learning paradigm, multilabel learning is applied in a variety of fields [3, 4], but it mainly focuses on an instance of the corresponding related or unrelated label.

Though multilabel learning can solve many ambiguity problems, it is not well-suited to some practical problems [5, 6]. For example, consider the image recognition problem of a natural scene that is annotated with mostly water, lots of sky, some cloud, a little land, and a few trees. As can be seen from Figure 1, each label in Figure 1(a) should be assigned a different importance. Multilabel learning mainly focuses on an instance of the corresponding related label or unrelated label, rather than the difference in importance [7]. This leads to the question of how to determine the importance of different labels in an instance. Label distribution learning (LDL) can reflect the importance of each label in an instance in a similar way to a probability distribution. Figure 1(b) shows an example of a label distribution. This scenario is encountered in many types of multilabel tasks, such as age estimation [7], expression recognition [8], and the prediction of crowd opinions [9].

In contrast to the multilabel learning output of a set of labels, the output of LDL is a probability distribution [10]. In recent years, LDL has become a popular topic of research as a new paradigm in machine learning. For instance, Geng et al. proposed the IIS-LDL and CPNN algorithms to estimate the ages of different faces [11]. Their approach achieves better results than previous age estimation algorithms, because they use more information in the training process. Thereafter, Geng developed a complete framework for LDL [10]. This framework not only defines LDL but also generalizes LDL algorithms and gives corresponding metrics to measure their performance.

At present, the parameter model for LDL is mainly based on Kullback-Leibler divergence [12]. Different models can be used to train the parameters, such as maximum entropy [13] or logistic regression [14], although there is no particular evidence to support their use. To some extent,



(a) Multilabel (b) Label distribution

Figure 1: A natural scene image which has been annotated with water, sky, cloud, land, and trees.

the LDL process ignores the linear relationship between the features and the label distribution. Unlike other applications, LDL aims to predict the label distribution rather than the category. Thus, the overall label distribution can be effectively reconstructed from the corresponding samples.

In this paper, we propose an LDL method that uses the property of sample self-representation to reconstruct the labels. As the labels are similar to a probability distribution, but not actually a probability distribution, in LDL we can represent the labels through the feature matrix instead of the distance between two probability distributions. With the above considerations, we use a least-squares model to establish the objective function. That is, as far as possible, each label distribution is represented as the linear combination of its relevant features. The goal of this optimization model is to minimize the residuals. We combine LDL with sparsity regularization to optimize the model and then introduce regularization terms to solve the model. To solve the objective function efficiently, we use the L_2-norm and $L_{2,1}$-norm as the regularization terms. The corresponding algorithms are named regularized sample self-representation RSSR-LDL2 and RSSR-LDL21. The proposed algorithms not only have strong interpretability but also avoid the problem of overfitting. In a series of experiments, we demonstrate that the similarity and distance in a variety of evaluation metrics are superior to those of four state-of-the-art algorithms. The results of the experimental analysis on public datasets show that the proposed method can effectively predict the labels.

The remainder of this paper is organized as follows. A brief review of related work on LDL and sparsity regularization is presented in Section 2. In Section 3, we introduce the LDL task and evaluation metrics. We describe the RSSR-LDL method and develop two algorithms in Section 4. Section 5 presents and analyzes the experimental results. Finally, we conclude this paper and present some ideas for future work in Section 6.

2. Related Work

The continued efforts of researchers have led to various LDL algorithms being proposed [9, 13, 15]. There are three main design strategies in the literature [10]: problem transformation, algorithm adaptation, and specialized algorithm design. Problem transformation (PT) takes the label distribution instances and transforms them into multilabel instances or single-label instances; PT-SVM and PT-Bayes are the representative algorithms in this class. Algorithm adaptation (AA) extends some existing supervised learning algorithms to deal with the problem of label distribution, such as the AA-kNN and AA-BP algorithms [10]. Unlike problem transformation and algorithm adaptation, specialized algorithm (SA) design sets up a direct model for the label distribution data. Typical algorithms include LDLogitBoost, based on logistic regression [16], SA-BFGS, based on maximum entropy [10], and DLDL, which combines LDL with deep learning [17, 18].

Unlike traditional clustering [19] or classification learning [20], the labels in LDL have similar patterns to probability distributions. According to the definition of the label distribution, we assume that there may be a function that matches the feature to the label. We find that each label can be well approximated by a linear combination of its relevant features.

Linear reconstruction is not strictly expressed as $y = \mathbf{A}x$, which $\mathbf{A}$ is a coefficient matrix. Then we transform it to minimum residual and optimize it by least-square method. In order to avoid overfitting and to solve the problem, regularization term L_2-norm is often added. In this way, the label distribution is constructed directly from the sample information of the data by the coefficient matrix. To find the corresponding relevant features while avoiding effects of noise in high dimensional data [21], we introduce sparse reconstruction [22, 23]. Sparsity reconstruction is the addition of sparse regularization terms L_1-norm or L_0-norm on the linear reconstruction. Nowadays, there are sparse

regularization terms $L_{2,1}$-norm and $L_{2,0}$-norm proposed gradually [24]. $L_{2,1}$-norm regularization is performed to select features across all of the data points with joint sparsity [25]. For matrix $\mathbf{A} = (a_{ij})$,

$$\|\mathbf{A}\|_{2,1} = \sum_i \sqrt{\sum_j a_{ij}^2}. \tag{1}$$

Sparsity reconstruction is widely used in machine learning, especially for data dimensionality reduction [26, 27]. For example, Cai proposed the MCFS algorithm by using L_1-regularized least-squares to deal with multicluster data [28], and Zhu et al. proposed the RMR algorithm based on regularized self-representation for feature selection [29]. Nie developed the RFS and JELSR algorithms using $L_{2,1}$-regularized least-squares to optimize the objective function [25, 30]. Furthermore, L_1-SVM and sparse logistic regression [31] have been shown to be effective. In general, linear reconstruction and sparsity reconstruction produce good performance in feature selection and classifier [32, 33]. Next, we will propose a label distribution learning method based on linear reconfiguration and sparsity reconstruction.

3. The Proposed Model

The goal of LDL is to obtain a set of probability distributions. Therefore, LDL is different from previous approaches in terms of the problem statement. In this section, the problem statement is briefly reviewed and the proposed model is introduced.

LDL is the process of describing an instance $\mathbf{x}$ more naturally using labels [10]. We assign a value d_x^y to each of the corresponding possible labels $\mathbf{y}$. This value represents the extent to which the label $\mathbf{y}$ describes instance $\mathbf{x}$, that is, the description degree. Taking account of the corresponding subset of labels can give a complete description of the sample. Therefore, it is assumed that $d_x^y \in [0, 1]$ and $\sum_y d_x^y = 1$. That is, the data for d_x^y have a form that is similar to a probability distribution for an instance. The learning process based on such data is called label distribution learning.

We use $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ to represent the n instances, $\mathbf{x}_i \in \mathbb{R}^d$, and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$. Let $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_c]$ denote the complete class labels, where $\mathbf{y}_k \in \mathbb{R}^c$ and $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_c$ represent the c labels. Corresponding label distribution $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n]^T$, among $\mathbf{d}_i = [\mathbf{d}_{\mathbf{x}_i}^{\mathbf{y}_1}, \mathbf{d}_{\mathbf{x}_i}^{\mathbf{y}_2}, \ldots, \mathbf{d}_{\mathbf{x}_i}^{\mathbf{y}_c}]$, represents the label distribution of instance $\mathbf{x}_i$. Concretely, $\mathbf{d}_{\mathbf{x}_i}^{\mathbf{y}_k}$ represents the extent to which the label $\mathbf{y}_k$ describes the instance $\mathbf{x}_i$. Therefore, we can represent the training set of label distribution learning $\mathbf{S} = [(\mathbf{x}_1, \mathbf{d}_1), (\mathbf{x}_2, \mathbf{d}_2), \ldots, (\mathbf{x}_n, \mathbf{d}_n)]^T$ in this paper. In addition, the test data is defined as $\mathbf{X}' = [\mathbf{x}_1', \mathbf{x}_2', \ldots, \mathbf{x}_m']^T$ and the corresponding predicted label distribution is defined as $\mathbf{D}' = [\mathbf{d}_1', \mathbf{d}_2', \ldots, \mathbf{d}_m']^T$.

We combine LDL with regularized sample self-representation to give the RSSR-LDL model. For RSSR-LDL, each sample and the corresponding description degree have the following relationship:

$$\mathbf{d}_i = \mathbf{x}_i \mathbf{P}, \tag{2}$$

where $\mathbf{P}$ is the transformation matrix from the sample to the description degree and $\mathbf{P} \in \mathbb{R}^{d \times c}$. According to the definition, this is equivalent to

$$\mathbf{D} = \mathbf{X}\mathbf{P}. \tag{3}$$

In general, $n > m$ for $\mathbf{X}$, and (3) cannot be solved [34].

In order to solve the optimal $\mathbf{P}$, we introduce residual sum function L:

$$L(\mathbf{P}) = \|\mathbf{X}\mathbf{P} - \mathbf{D}\|_2^2. \tag{4}$$

When $\mathbf{P} = \widehat{\mathbf{P}}$, the minimum value of $L(\mathbf{P})$, the objective function of the model is

$$\widehat{\mathbf{P}} = \arg\min_{\mathbf{P}} \|\mathbf{X}\mathbf{P} - \mathbf{D}\|_2^2. \tag{5}$$

Using the difference values from (4), we obtain

$$\mathbf{X}^T\mathbf{X}\mathbf{P} - \mathbf{X}^T\mathbf{D} = 0. \tag{6}$$

When $\mathbf{X}$ is not full rank, or there is a significant linear correlation between columns, the determinant will be close to 0, which makes the calculation of $\mathbf{X}^T\mathbf{X}$ an ill-posed problem. This will introduce a large error into the calculation of $(\mathbf{X}^T\mathbf{X})^{-1}$, resulting in a lack of stability and reliability in (5). Therefore, we introduce a regularization term $R(\mathbf{P})$ with parameter γ to optimize the objective function.

In other words,

$$\widehat{\mathbf{P}} = \arg\min_{\mathbf{P}} \|\mathbf{X}\mathbf{P} - \mathbf{D}\|_2^2 + \gamma R(\mathbf{P}), \quad \gamma > 0. \tag{7}$$

There are several possible regularizations [25],

$$\begin{aligned} R_1(\mathbf{P}) &= \sum_{j=1}^{c} \left\|\mathbf{p}_j\right\|_1, \\ R_2(\mathbf{P}) &= \|\mathbf{P}\|_2^2, \\ R_3(\mathbf{P}) &= \sum_{k=1}^{d} \sqrt{\sum_{j=1}^{c} \mathbf{P}_{kj}^2}; \end{aligned} \tag{8}$$

$R_1(\mathbf{P})$ is the LASSO regularization. $R_2(\mathbf{P})$ is the ridge regression, also known as Tikhonov regularization. It is the most frequently used regularization method for ill-posed problem. $R_3(\mathbf{P})$ is a new joint regularization [25, 29].

4. Regularized Model and Algorithm

To solve the objective function efficiently, we use the L_2-norm and $L_{2,1}$-norm to regularize the RSSR-LDL model, resulting in RSSR-LDL2 and RSSR-LDL21. The RSSR-LDL2 and RSSR-LDL21 algorithms are presented in this section.

4.1. Regularized Sample Self-Representation by L_2-Norm. We use the L_2-norm of $\mathbf{P}$ to solve the RSSR-LDL problem. For convenience, we use the regularization term $R_2(\mathbf{P})$ in (7). Then, (7) is as follows:

$$\widehat{\mathbf{P}} = \arg\min_{\mathbf{P}} \|\mathbf{X}\mathbf{P} - \mathbf{D}\|_2^2 + \gamma_1 \|\mathbf{P}\|_2^2. \tag{9}$$

Input: Train matrix $\mathbf{S}$, test data $\mathbf{X}'$ and the regularization parameter γ_1.
Output: The corresponding predicted label distribution of test data $\mathbf{X}', \mathbf{d}'_1, \mathbf{d}'_2, \ldots, \mathbf{d}'_m$.
(1) $\mathbf{P} = (\mathbf{X}^T\mathbf{X} + \gamma_1\mathbf{I})^{-1}\mathbf{X}^T\mathbf{D}$ where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ and $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_n]^T$; // Computational transformation matrix.
(2) $\mathbf{d}_i^0 = \mathbf{x}_i'\mathbf{P}$; // Using the transition matrix to represent the distribution of the predicted labels.
(3) $\mathbf{d}_i' = |\mathbf{d}_i^0| / \sum_{k=1}^{c} |\mathbf{d}_i^0|$; // We normalize $\mathbf{d}_i^0$ $(i = 1, 2, \ldots, m)$ because of $d_x^y \in [0, 1]$ and $\sum_y d_x^y = 1$.

Algorithm 1: Regularized sample self-representation by L_2-norm (RSSR-LDL2).

Input: Train matrix $\mathbf{S}$, test data $\mathbf{X}'$, the number of iterations **Iter** and the regularization parameter γ_2.
Output: The corresponding predicted label distribution of test data $\mathbf{X}', \mathbf{d}'_1, \mathbf{d}'_2, \ldots, \mathbf{d}'_m$.
(1) Initialize $\mathbf{B}_0 \in \mathbb{R}^{(d+n)\times(d+n)}$ which is an identity matrix, set $t = 0$; // Initialization
(2) **for** $t = 1$ to **Iter do**.
(3) $\mathbf{Q}_{t+1} = \mathbf{B}_t^{-1}\mathbf{Z}^{\mathbf{T}}(\mathbf{Z}\mathbf{B}_t^{-1}\mathbf{Z}^T)^{-1}\mathbf{D}$, // calculate $\mathbf{Q}_{t+1}$
(4) $\mathbf{B}_{t+1} = \mathrm{diag}(1/2\mathbf{q}_{\mathbf{t+1}}^{\mathbf{l}})$, // calculate $\mathbf{B}_{t+1}$
(5) **end for**
(6) $\mathbf{P} = \mathbf{Q}(d, :)$; // Removing the matrix $\mathbf{A}$ part of matrix $\mathbf{Q}$.
(7) $\mathbf{d}_i^0 = \mathbf{x}_i'\mathbf{P}$; // Using the transition matrix to represent the distribution of the predicted labels.
(8) $\mathbf{d}_j' = |\mathbf{d}_i^0| / \sum_{k=1}^{c} |\mathbf{d}_i^0|$; // Because of $d_x^y \in [0, 1]$ and $\sum_y d_x^y = 1$, the normalized $\mathbf{d}_i^0$, $i = 1, 2, \ldots, m$.

Algorithm 2: Regularized sample self-representation by $L_{2,1}$-norm (RSSR-LDL21).

Because (9) is smooth, (9) can be solved by the differential as (10).

$$\mathbf{X}^T\mathbf{X}\mathbf{P} + \gamma_1\mathbf{I}\mathbf{P} - \mathbf{X}^T\mathbf{D} = 0, \tag{10}$$

or equivalently,

$$\left(\mathbf{X}^T\mathbf{X} + \gamma_1\mathbf{I}\right)\mathbf{P} = \mathbf{X}^T\mathbf{D}. \tag{11}$$

$\mathbf{X}^T\mathbf{X} + \gamma_1\mathbf{I}$ is a nonsingular matrix definitely; then

$$\mathbf{P} = \left(\mathbf{X}^T\mathbf{X} + \gamma_1\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{D}. \tag{12}$$

We can predict the label distribution of the test dataset using the learned matrix $\mathbf{P}$. Specifically, the predictive label distribution is defined as follows:

$$\begin{aligned} \mathbf{d}_i^0 &= \mathbf{x}_i'\mathbf{P}, \\ \mathbf{d}_i' &= \frac{\left|\mathbf{d}_i^0\right|}{\sum_{k=1}^{c}\left|\mathbf{d}_i^0\right|}. \end{aligned} \tag{13}$$

Borrowing from the above theoretical analysis, we summarize Algorithm 1.

Algorithm 1 does not include an iterative process. In other words, the matrix $\mathbf{P}$ can be solved directly, which makes the algorithm faster. Although this approach is efficient and easy to understand, it is not very accurate. Thus, in the next section, we use the $L_{2,1}$-norm of $\mathbf{P}$ to solve the RSSR-LDL problem.

4.2. Regularized Sample Self-Representation by $L_{2,1}$-Norm. Combined with the characteristics of $R_1(\mathbf{P})$ and $R_2(\mathbf{P})$, we choose the $L_{2,1}$-norm of $\mathbf{P}$,that is, $R_3(\mathbf{P})$, as the regularization term. This gives the RSSR-LDL21 algorithm. The objective optimization function of (7) is shown in the following expression:

$$\widehat{\mathbf{P}} = \arg\min_{\mathbf{P}} \quad \|\mathbf{X}\mathbf{P} - \mathbf{D}\|_{2,1} + \gamma_2 \|\mathbf{P}\|_{2,1}, \tag{14}$$

which can be transformed into

$$\arg\min_{\mathbf{P}} \quad \frac{1}{\gamma_2}\|\mathbf{X}\mathbf{P} - \mathbf{D}\|_{2,1} + \|\mathbf{P}\|_{2,1}. \tag{15}$$

According to [25], this can be further transformed into

$$\begin{aligned} \min_{\mathbf{Q}} \quad & \|\mathbf{Q}\|_{2,1} \\ \text{s.t.} \quad & \mathbf{Z}\mathbf{Q} = \mathbf{D}, \end{aligned} \tag{16}$$

where $\mathbf{Q} = [\mathbf{P}, \mathbf{A}]^T \in \mathbb{R}^{(d+n)\times c}$, $\mathbf{A} \in \mathbb{R}^{n\times c}$, $\mathbf{Z} = [\mathbf{X}^T, \gamma_2\mathbf{I}] \in \mathbb{R}^{n\times(d+n)}$, and $\mathbf{I} \in \mathbb{R}^{n\times n}$. The problem in (16) becomes one of solving a Lagrangian function; that is,

$$\mathbf{Q} = \mathbf{B}^{-1}\mathbf{Z}^{\mathbf{T}}\left(\mathbf{Z}\mathbf{B}^{-1}\mathbf{Z}^T\right)^{-1}\mathbf{D}, \tag{17}$$

where $\mathbf{B} \in \mathbb{R}^{(d+n)\times(d+n)}$ is a diagonal matrix with $d_{ll} = 1/2\|\mathbf{q}^{\mathbf{l}}\|_2$, $l = 1, 2, \ldots, d+n$. The solution in (17) is convergent [25], so the iteration is viable. The RSSR-LDL21 algorithm is shown in Algorithm 2.

In Algorithm 2, the iteration is repeated until **Iter** = 30. In each iteration, $\mathbf{Q}$ is calculated with the previous $\mathbf{B}$ and $\mathbf{B}$ is calculated with the current $\mathbf{Q}$.

5. Experiments

To demonstrate the performance of the proposed RSSR-LDL2 and RSSR-LDL21 algorithms, we apply them to gene expression

Table 1: Evaluation metrics description.

ID	Evaluation metrics	Expression
1	Chebyshev ↓	$\text{distance}_1 = \max_k \left\|\mathbf{d}_k - \mathbf{d}'_k\right\|$
2	Clark ↓	$\text{distance}_2 = \sqrt{\sum_{k=1}^{m} \frac{\left(\mathbf{d}_k - \mathbf{d}'_k\right)^2}{\left(\mathbf{d}_k + \mathbf{d}'_k\right)^2}}$
3	Canberra ↓	$\text{distance}_3 = \sum_{k=1}^{m} \frac{\left\|\mathbf{d}_k - \mathbf{d}'_k\right\|}{\mathbf{d}_k + \mathbf{d}'_k}$
4	Intersection ↑	$\text{similarity}_1 = \sum_{k=1}^{m} \min\left(\mathbf{d}_k, \mathbf{d}'_k\right)$
5	Cosine ↑	$\text{similarity}_2 = \frac{\sum_{k=1}^{m} \mathbf{d}_k \mathbf{d}'_k}{\sqrt{\sum_{k=1}^{c} \mathbf{d}_k^2} \sqrt{\sum_{k=1}^{m} \left(\mathbf{d}'_k\right)^2}}$

Table 2: Data description.

ID	Dataset	♯ Instance	♯ Feature	♯ Label	♯ Data type
1	Movie	7755	1869	5	Movie score
2	SBU-3DFE	2500	243	6	Facial expression
3	SJAFFE	213	243	6	Facial expression
4	Yeast-alpha	2465	24	18	Gene expression
5	Yeast-cdc	2465	24	15	Gene expression
6	Yeast-cold	2465	24	4	Gene expression
7	Yeast-diau	2465	24	7	Gene expression
8	Yeast-dtt	2465	24	4	Gene expression
9	Yeast-elu	2465	24	14	Gene expression
10	Yeast-heat	2465	24	6	Gene expression
11	Yeast-spo	2465	24	6	Gene expression
12	Yeast-spo5	2465	24	3	Gene expression

levels, facial expression, and movie score problems. In this section, we use five evaluation metrics to test the proposed algorithms on 12 publicly available datasets (http://cse.seu.edu.cn/PersonalPage/xgeng/LDL/index.htm). The proposed algorithms are also compared with four state-of-the-art LDL algorithms.

5.1. Evaluation Metrics. In LDL, there are multiple labels associated with each instance, and these reflect the importance of each label for the instance. As a result, performance evaluation is different from that of both single- and multilabel learning. Because the label distribution is similar to a probability distribution, we use the similarity and distance between the original distribution and the predicted distribution to evaluate the effectiveness of LDL algorithms. There are many measures of the distance and similarity between probability distributions. In [35], 41 kinds of distance and similarity evaluation metrics were identified across eight classes. The various distance/similarity measures offer different performances in terms of comparing two probability distributions.

According to the agglomerative single linkage with average clustering method [36], screening rules [10], and experimental conditions, we evaluated five methods: the Chebyshev distance [37], Clark distance [38], Canberra distance [39], intersection similarity [36], and cosine similarity [39]. The related names and expressions are listed in Table 1. A "↓" after the distance measure indicates that smaller values are better, whereas "↑" after the similarity measure indicates that larger values are better.

5.2. Experimental Setting. Experiments are conducted on 12 public datasets. In the movie dataset, each instance represents the characteristics of a movie and the category score that the movie may belong to. SBU-3DFE and SJAFFE datasets represent facial expression images. Each instance represents a facial expression and scores of possible expression class. The Yeast family contains nine yeast gene expression levels. Each instance represents the expression level of a gene at a certain time. These datasets are described in Table 2.

To verify the effectiveness and performance of our LDL method, we compared the RSSR-LDL2 and RSSR-LDL21 algorithms with four existing LDL algorithms. According to [10], we selected comparative algorithms that use different strategies.

PT-SVM. PT-SVM is applied to training sets in which label distribution is obtained by the problem resampling method [40]. PT-SVM uses pairwise coupling to solve the multiclassification problem [41]. This algorithm calculates the posterior probability of each class as the description degree of a label.

AA-BP. AA-BP is a three-layer backpropagation neural network. This algorithm has n input units and c output units, which receive $\mathbf{X}$ and output $\mathbf{D}$, respectively.

Table 3: *Chebyshev distance* ↓ (mean ± std) × 10^3 of different algorithms on the twelve datasets. The best results are enlightened in bold and the second best results are italicized.

Algorithms	PT-SVM	AA-BP	SA-IIS	SA-BFGS	RSSR-LDL2	RSSR-LDL21
Movie	233.5 ± 26.0	139.8 ± 1.4	129.7 ± 3.0	126.6 ± 3.6	**113.4 ± 0.7**	*114.1 ± 3.6*
SBU-3DFE	142.2 ± 5.4	144.2 ± 6.1	133.2 ± 4.7	**104.2 ± 4.5**	124.0 ± 4.9	*107.6 ± 2.4*
SJAFFE	121.0 ± 10.3	136.3 ± 16.5	117.2 ± 8.5	105.2 ± 15.0	*96.3 ± 18.9*	**90.7 ± 9.3**
Yeast-alpha	13.8 ± 0.4	37.6 ± 2.4	16.9 ± 0.3	**13.4 ± 0.4**	**13.4 ± 0.4**	*13.4 ± 0.5*
Yeast-cdc	17.2 ± 0.8	38.0 ± 1.9	20.0 ± 0.5	**16.2 ± 0.4**	*16.2 ± 0.5*	*16.2 ± 0.5*
Yeast-cold	57.8 ± 3.9	57.8 ± 2.2	56.7 ± 1.9	51.1 ± 2.2	*51.0 ± 1.7*	**50.9 ± 1.9**
Yeast-diau	43.2 ± 3.9	49.1 ± 1.9	41.2 ± 1.3	*36.9 ± 1.1*	**36.9 ± 1.0**	36.9 ± 1.3
Yeast-dtt	38.6 ± 2.2	44.6 ± 2.5	43.3 ± 1.6	36.0 ± 1.8	**35.9 ± 1.5**	*35.9 ± 1.7*
Yeast-elu	17.0 ± 0.3	37.8 ± 2.5	20.2 ± 0.8	*16.3 ± 0.5*	**16.2 ± 0.5**	**16.2 ± 0.5**
Yeast-heat	44.0 ± 1.0	55.1 ± 4.0	46.5 ± 1.1	42.3 ± 1.0	*42.2 ± 1.5*	**42.2 ± 1.1**
Yeast-spo	64.7 ± 3.0	66.3 ± 2.9	61.7 ± 1.7	58.3 ± 2.8	*58.2 ± 2.7*	**58.0 ± 2.2**
Yeast-spo5	92.9 ± 3.9	95.7 ± 4.2	94.6 ± 2.4	91.4 ± 3.1	**91.1 ± 3.9**	*91.2 ± 3.4*

SA-IIS. SA-IIS uses the maximum entropy to solve the LDL problem. The optimization strategy of this algorithm is similar to that of the scaling-based IIS [42].

SA-BFGS. SA-BFGS is an improved algorithm based on SA-IIS. This improved algorithm employs an effective quasi-Newton method and is more efficient than the standard line search approach.

For the parameter settings in these four algorithms, we refer to [10]. For our algorithms, we tuned the regularization parameter using values of $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ and present the best results [29]. The performance of the above LDL algorithms was evaluated by considering the distance and similarity between the original label distribution and the predicted label distribution.

5.3. Results Analysis of Experiments. In this section, the performance of the proposed algorithms is compared with that of four existing state-of-the-art LDL algorithms in terms of five evaluation metrics. We also present the predicted label distribution given by the six algorithms and the real label distribution.

5.3.1. Distance and Similarity Comparison. To verify the advantages of the proposed RSSR-LDL2 and RSSR-LDL21, experiments were conducted on 12 public datasets. Each experiment used tenfold cross-validation [43, 44], and the mean value and standard deviation of each evaluation were recorded. Because many results were close to zero, they are represented as "(mean ± std) × 10^3." The main measure of the size of individual differences is the distance, whereas the similarity reflects the trend and direction of the vector. Therefore, we use distance and similarity to demonstrate the superiority of the proposed algorithms.

The results for the Chebyshev distance, Clark distance, Canberra metric, cosine coefficient, and intersection similarity are presented in Tables 3–6, respectively. In each table, the best results are given in bold and the second-best results are italicized (if the mean is the same, the algorithm with the smaller standard deviation is considered to be better). The first evaluation metrics measure distance, and so smaller values are better; the latter two measure similarity, and so larger values are better. From these results, we can see that RSSR-LDL21 achieves the best performance of all the algorithms and RSSR-LDL2 is better than the others.

From the results in Tables 4–6, our algorithms have obvious advantages. In particular, the L21 algorithm offers better performance than the other algorithms with almost every dataset. The SA-BFGS algorithm achieves equivalent performance in terms of the Chebyshev distance (Table 3) and cosine coefficient (Table 7) with some datasets, mainly those for the yeast genes. In addition, our algorithms not only produce good results, but they are also very stable, especially RSSR-LDL21.

The proposed algorithms perform differently with the different datasets. The results show that the RSSR-LDL approach has an absolute advantage over the other algorithms with the movie dataset. This is because the characteristics of sparse representation offer obvious advantages when there are a large number of features. The proposed algorithms continue to offer some advantages over the other algorithms with the facial expression datasets, although some results are similar to those given by the SA-BFGS algorithm. As the number of features in the yeast datasets is small, our algorithms do not show the best performance with all evaluation metrics but still achieve similar performance to the SA-BFGS algorithm. Moreover, the performance of the proposed algorithms is better than the other comparative algorithms. Especially, there is a more obvious advantage in high dimensional data.

5.3.2. Label Distribution Showing. Unlike classification learning and clustering, LDL reflects the importance of each label for an instance. Hence, our ultimate goal is no longer categorization but a sort of probability distribution. Two typical examples of the original label distribution and that predicted by the six LDL algorithms are presented in Table 8. We select the $[n/2]$th sample of the label distribution as a demonstration.

In Table 8, the second and third columns represent the real label distribution and the predicted label distributions given by the six different algorithms for the movie and SBU-3DFE datasets, respectively. Each point represents the

Table 4: *Clark Distance* ↓ (mean ± std) × 10^3 of different algorithms on the twelve datasets. The best results are enlightened in bold and the second best results are italicized.

Algorithms	PT-SVM	AA-BP	SA-IIS	SA-BFGS	RSSR-LDL2	RSSR-LDL21
Movie	871.2 ± 77.4	643.8 ± 11.8	553.6 ± 12.9	551.8 ± 10.7	*521.4 ± 5.4*	**514.7 ± 10.6**
SBU-3DFE	430.1 ± 16.3	469.7 ± 25.2	410.0 ± 9.0	**348.5 ± 11.4**	390.2 ± 5.6	*380.0 ± 2.9*
SJAFFE	437.9 ± 26.8	508.0 ± 46.4	417.6 ± 18.1	420.0 ± 36.6	*366.9 ± 36.3*	**348.5 ± 18.1**
Yeast-alpha	220.9 ± 5.1	752.2 ± 54.1	260.4 ± 4.2	210.0 ± 6.7	*209.3 ± 5.3*	**209.2 ± 5.9**
Yeast-cdc	228.1 ± 8.9	585.1 ± 27.1	258.9 ± 4.6	215.8 ± 3.5	*215.1 ± 5.5*	**214.7 ± 5.7**
Yeast-cold	155.9 ± 10.0	156.9 ± 5.7	153.1 ± 5.7	139.5 ± 6.3	*139.3 ± 4.8*	**139.0 ± 5.7**
Yeast-diau	235.3 ± 20.4	270.7 ± 12.0	222.2 ± 7.0	200.5 ± 6.9	*200.3 ± 5.1*	**200.1 ± 6.2**
Yeast-dtt	104.7 ± 6.3	121.6 ± 6.5	116.3 ± 5.2	98.3 ± 5.5	*98.0 ± 3.9*	**97.9 ± 4.9**
Yeast-elu	210.9 ± 4.5	528.0 ± 40.0	240.5 ± 7.1	198.9 ± 5.8	*198.3 ± 5.3*	**198.3 ± 4.2**
Yeast-heat	190.3 ± 5.8	242.0 ± 20.4	200.5 ± 4.7	182.7 ± 5.0	*182.3 ± 6.2*	**182.0 ± 4.1**
Yeast-spo	272.4 ± 11.3	287.5 ± 12.5	263.7 ± 5.8	249.6 ± 11.9	*249.4 ± 11.4*	**248.7 ± 8.5**
Yeast-spo5	187.0 ± 8.6	192.1 ± 8.8	190.1 ± 4.8	184.3 ± 7.2	*183.7 ± 8.8*	**183.7 ± 6.9**

Table 5: *Canberra Meric* ↓ (mean ± std) × 10^3 of different algorithms on the twelve datasets. The best results are enlightened in bold and the second best results are italicized.

Algorithms	PT-SVM	AA-BP	SA-IIS	SA-BFGS	RSSR-LDL2	RSSR-LDL21
Movie	1693 ± 183.1	1232 ± 22.2	1063 ± 27.0	1063 ± 22.9	*992.0 ± 10.3*	**989.1 ± 24.6**
SBU-3DFE	925.7 ± 34.3	984.1 ± 47.6	888.8 ± 20.6	**725.1 ± 24.9**	836.2 ± 14.3	*782.5 ± 8.6*
SJAFFE	917.8 ± 59.2	1034.8 ± 97.7	870.6 ± 44.4	862.5 ± 76.1	*735.9 ± 80.7*	**705.8 ± 46.8**
Yeast-alpha	723.1 ± 19.3	2483.5 ± 172.3	859.2 ± 16.0	681.9 ± 21.1	*679.0 ± 16.7*	**678.3 ± 19.1**
Yeast-cdc	685.7 ± 24.7	1772.5 ± 74.3	786.7 ± 13.4	647.3 ± 14.9	*645.0 ± 14.6*	**642.3 ± 14.9**
Yeast-cold	269.5 ± 17.6	269.9 ± 8.6	264.5 ± 10.0	240.1 ± 10.0	*239.7 ± 8.6*	**239.4 ± 9.7**
Yeast-diau	508.8 ± 47.2	584.7 ± 28.5	480.8 ± 13.9	430.5 ± 15.4	*429.9 ± 10.6*	**429.7 ± 11.2**
Yeast-dtt	179.9 ± 10.1	209.4 ± 11.3	201.0 ± 8.8	169.0 ± 8.8	*168.6 ± 5.9*	**168.4 ± 8.4**
Yeast-elu	621.2 ± 16.2	1546.8 ± 120.2	714.7 ± 18.0	582.6 ± 18.0	*581.1 ± 11.5*	**581.0 ± 13.1**
Yeast-heat	380.7 ± 11.8	486.5 ± 40.4	403.3 ± 9.9	364.4 ± 9.0	*363.5 ± 12.0*	**362.8 ± 7.6**
Yeast-spo	562.8 ± 21.8	589.2 ± 24.6	541.6 ± 12.8	512.9 ± 24.1	*512.5 ± 23.4*	**511.8 ± 18.3**
Yeast-spo5	287.3 ± 12.8	295.5 ± 13.3	292.3 ± 7.3	283.1 ± 10.5	**282.1 ± 12.9**	*282.3 ± 10.5*

Table 6: *Intersection* ↑ (mean ± std) × 10^3 of different algorithms on the twelve datasets. The best results are enlightened in bold and the second best results are italicized.

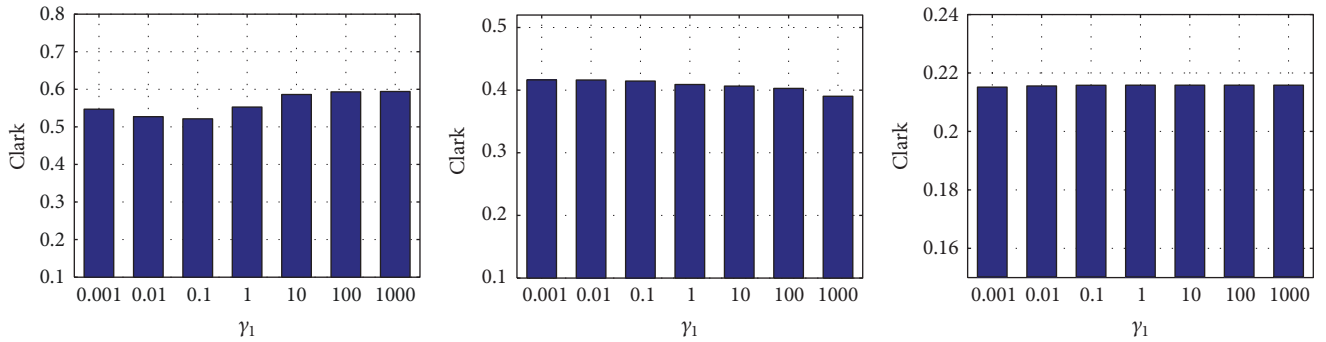
Algorithms	PT-SVM	AA-BP	SA-IIS	SA-BFGS	RSSR-LDL2	RSSR-LDL21
Movie	675.3 ± 45.9	795.9 ± 3.1	820.7 ± 4.1	822.1 ± 4.2	**837.9 ± 1.4**	*837.2 ± 5.0*
SBU-3DFE	833.8 ± 6.1	823.0 ± 7.9	840.8 ± 4.0	**871.4 ± 4.7**	850.4 ± 3.2	*864.3 ± 1.7*
SJAFFE	843.3 ± 10.5	823.1 ± 18.1	851.8 ± 8.5	858.1 ± 15.6	*878.4 ± 15.6*	**883.3 ± 9.2**
Yeast-alpha	960.1 ± 1.0	870.8 ± 8.1	952.0 ± 0.9	962.4 ± 1.1	*962.5 ± 0.9*	**962.6 ± 1.0**
Yeast-cdc	954.8 ± 1.6	888.2 ± 4.2	947.6 ± 0.9	957.4 ± 1.1	*957.6 ± 0.9*	**957.7 ± 1.0**
Yeast-cold	933.2 ± 4.5	933.4 ± 2.0	934.5 ± 2.4	940.8 ± 2.3	*940.9 ± 2.1*	**941.0 ± 2.3**
Yeast-diau	929.1 ± 6.7	919.0 ± 4.0	932.8 ± 1.9	*940.3 ± 2.1*	**940.4 ± 1.4**	**940.4 ± 1.4**
Yeast-dtt	955.6 ± 2.4	948.3 ± 2.8	950.1 ± 1.9	958.3 ± 2.0	**958.4 ± 1.4**	*958.4 ± 2.0*
Yeast-elu	956.1 ± 1.2	895.0 ± 7.8	948.9 ± 1.3	958.9 ± 1.2	**959.0 ± 0.8**	*959.0 ± 1.0*
Yeast-heat	937.4 ± 1.8	920.3 ± 6.2	933.4 ± 1.7	940.2 ± 1.3	*940.3 ± 1.9*	**940.4 ± 1.2**
Yeast-spo	906.7 ± 3.6	903.2 ± 3.8	910.5 ± 2.2	915.6 ± 3.9	*915.6 ± 3.7*	**915.8 ± 3.0**
Yeast-spo5	907.1 ± 3.9	904.3 ± 4.2	905.4 ± 2.4	908.6 ± 3.1	**908.9 ± 3.9**	*908.8 ± 3.4*

corresponding value of a label in the subgraph in Table 8, and the spline shows the trend in the label distribution. According to the distribution law of the midpoint of the graph, the movie distribution was fitted using a Gaussian function and SBU-3DFE was fitted with a smooth spline.

Table 8 indicates that the proposed algorithms achieve perfect performance. On the one hand, the RSSR-LDLL21 algorithm has an absolute advantage, with the value and trend being almost consistent with the real label distribution. On the other hand, the RSSR-LDL2 algorithm is not as good as RSSR-LDL21 but achieves the same performance as SA-BFGS, which is obviously better than the other three comparative algorithms in terms of distance and similarity.

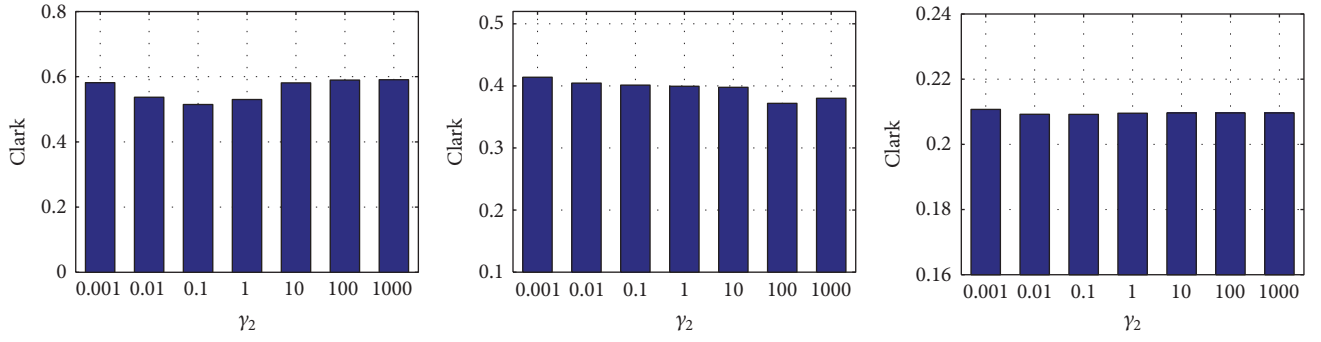
Table 7: *Cosine* ↑ (mean ± std) × 10^3 of different algorithms on the twelve datasets. The best results are enlightened in bold and the second best results are italicized.

Algorithms	PT-SVM	AA-BP	SA-IIS	SA-BFGS	RSSR-LDL2	RSSR-LDL21
Movie	766.2 ± 53.3	901.0 ± 2.4	922.5 ± 3.1	923.0 ± 3.6	**936.9 ± 0.6**	*934.4 ± 4.7*
SBU-3DFE	912.9 ± 5.4	902.6 ± 8.2	921.8 ± 3.4	**947.2 ± 3.7**	931.8 ± 3.6	*943.4 ± 1.6*
SJAFFE	927.3 ± 8.9	902.8 ± 20.6	934.3 ± 7.1	939.8 ± 13.9	*953.8 ± 15.7*	**957.9 ± 7.8**
Yeast-alpha	*994.1 ± 0.3*	945.1 ± 6.0	991.5 ± 0.2	**994.6 ± 0.3**	**994.6 ± 0.3**	**994.6 ± 0.3**
Yeast-cdc	992.5 ± 0.5	957.5 ± 3.2	990.2 ± 0.4	**993.3 ± 0.2**	*993.3 ± 0.3*	*993.3 ± 0.3*
Yeast-cold	985.5 ± 1.9	985.6 ± 1.2	986.1 ± 1.0	*988.6 ± 0.9*	**988.6 ± 0.8**	988.6 ± 1.0
Yeast-diau	983.8 ± 2.6	978.2 ± 2.0	985.1 ± 0.8	*988.0 ± 0.8*	**988.0 ± 0.6**	**988.0 ± 0.6**
Yeast-dtt	993.4 ± 0.8	991.1 ± 0.9	991.6 ± 0.7	*994.1 ± 0.7*	**994.1 ± 0.4**	*994.1 ± 0.7*
Yeast-elu	993.4 ± 0.3	961.8 ± 5.0	990.9 ± 0.6	*994.0 ± 0.3*	**994.1 ± 0.3**	**994.1 ± 0.3**
Yeast-heat	986.8 ± 0.6	978.9 ± 3.3	985.4 ± 0.7	**988.0 ± 0.5**	*988.0 ± 0.8*	**988.0 ± 0.5**
Yeast-spo	971.3 ± 2.1	970.2 ± 2.2	974.5 ± 1.2	*977.0 ± 2.0*	**977.0 ± 1.7**	**977.0 ± 1.7**
Yeast-spo5	973.1 ± 2.1	971.4 ± 2.7	972.3 ± 1.1	974.1 ± 1.6	**974.3 ± 2.0**	*974.2 ± 1.5*



(a) Movie (b) SBU-3DFE (c) Yeast-alpha

Figure 2: Clark distance of RSSR-LDL2 with respect to γ_1.



(a) Movie (b) SBU-3DFE (c) Yeast-alpha

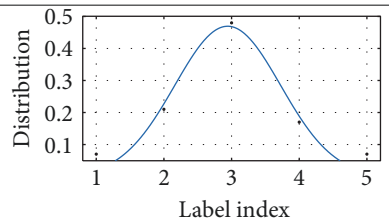
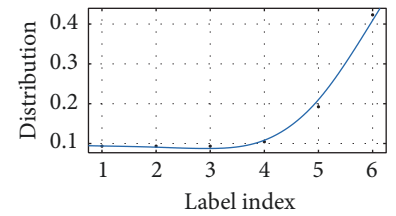
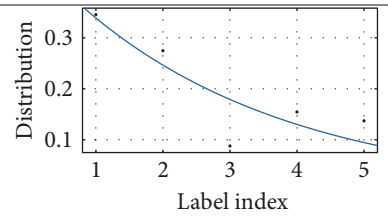
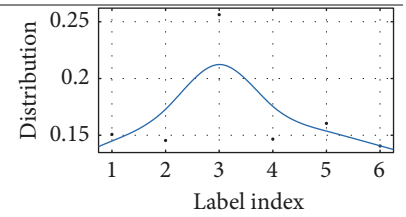
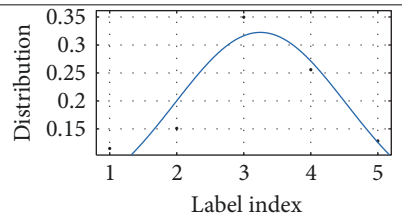
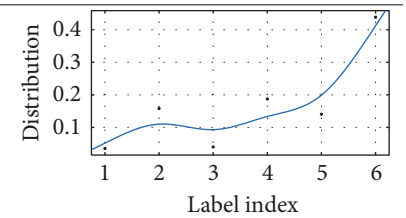
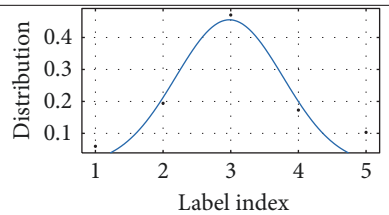
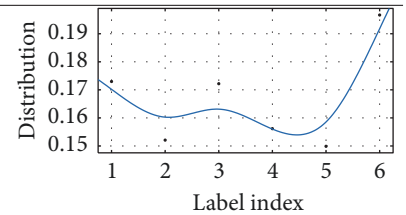
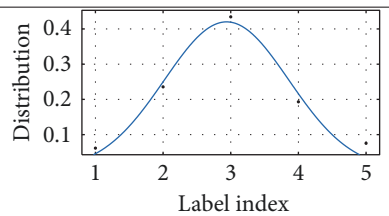
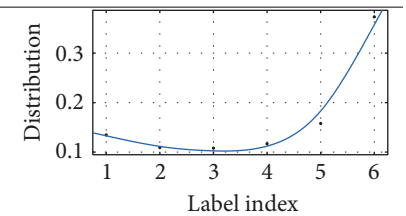
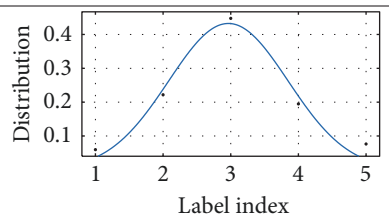
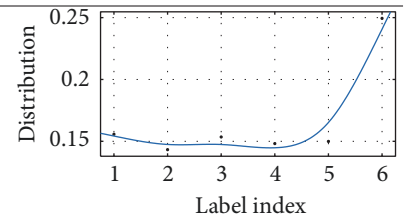
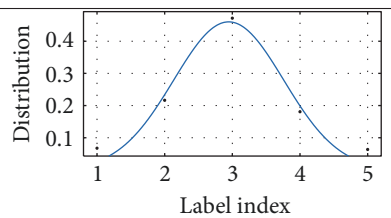
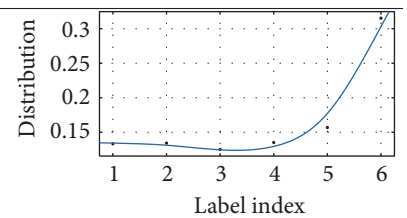
Figure 3: Clark distance of RSSR-LDL21with respect to γ_2.

5.3.3. Parameter Sensitivity. Like many other learning algorithms, RSSR-LDL has parameters that must be tuned in advance. We tuned $\gamma_1 = \gamma_2 = \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ and then recorded the best results given in Tables 3–8. For RSSR-LDL2, the Clark distance given by $\gamma_1 = \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ on 3 representative datasets is shown in Figure 2 which belongs to three different data types, respectively. We observe that RSSR-LDL2 is relatively insensitive to γ_1 for the facial expression and gene expression datasets, whereas it is slightly more sensitive for movie score datasets. Interestingly, in Figure 3, note that γ_2 in RSSR-LDL21 is similar to γ_1.

6. Conclusion and Future Work

LDL deals with instances associated with multiple labels but also reflects the importance degree of each label on the instance. In this paper, we proposed a new criterion

Table 8: The real and predictive distribution of two typical examples on six algorithms.

Algorithms	Movie	SBU-3DFE
Real		
PT-SVM		
AA-BP		
SA-IIS		
SA-BFGS		
RSSR-LDL2		
RSSR-LDL21		

for LDL using regularized sample self-representation. We reconstructed the labels with features and a transformation matrix and described each label as a linear combination of features. Then, we used the L_2-norm and $L_{2,1}$-norm as regularization terms to optimize the transformation matrix. We conducted experiments on 12 real datasets and compared the proposed algorithms with four existing LDL algorithms using five evaluation metrics. The experimental results show that the proposed algorithms are efficient and accurate. In future work, we will use a least-angle regression model to develop a better generalization model for solving practical problems.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is in part supported by National Science Foundation of China (under Grant nos. 61379049, 61379089, and 61703196).

References

[1] M.-L. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-2010*, pp. 999–1007, USA, July 2010.

[2] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[3] Z.-H. Zhou and M.-L. Zhang, "Multi-instance multi-label learning with application to scene classification," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS '06)*, pp. 1609–1616, December 2006.

[4] M.-L. Zhang and L. Wu, "LIFT: multi-label learning with label-specific features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 107–120, 2015.

[5] Y.-K. Li, M.-L. Zhang, and X. Geng, "Leveraging implicit relative labeling-importance information for effective multi-label learning," in *Proceedings of the 15th IEEE International Conference on Data Mining, ICDM 2015*, pp. 251–260, USA, November 2015.

[6] W. Zhu, "Relationship between generalized rough sets based on binary relation and covering," *Information Sciences*, vol. 179, no. 3, pp. 210–225, 2009.

[7] Z. He, X. Li, Z. Zhang et al., "Data-dependent label distribution learning for age estimation," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3846–3858, 2017.

[8] Y. Zhou, H. Xue, and X. Geng, "Emotion distribution recognition from facial expressions," in *Proceedings of the 23rd ACM International Conference on Multimedia, MM 2015*, pp. 1247–1250, Australia, October 2015.

[9] X. Geng and P. Hou, "Pre-release prediction of crowd opinion on movies by label distribution learning," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI 2015*, pp. 3511–3517, arg, July 2015.

[10] X. Geng, "Label Distribution Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1734–1748, 2016.

[11] X. Geng, C. Yin, and Z.-H. Zhou, "Facial age estimation by learning from label distributions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2401–2412, 2013.

[12] T. van Erven and P. Harremo, "Rényi divergence and Kullback-Leibler divergence," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 60, no. 7, pp. 3797–3820, 2014.

[13] X. Geng, Q. Wang, and Y. Xia, "Facial age estimation by adaptive label distribution learning," in *Proceedings of the 22nd International Conference on Pattern Recognition, ICPR 2014*, pp. 4465–4470, Sweden, August 2014.

[14] J. Pearce and S. Ferrier, "Evaluating the predictive performance of habitat models developed using logistic regression," *Ecological Modelling*, vol. 133, no. 3, pp. 225–245, 2000.

[15] Z. Zhang, M. Wang, and X. Geng, "Crowd counting in public video surveillance by label distribution learning," *Neurocomputing*, vol. 166, pp. 151–163, 2015.

[16] C. Xing, X. Geng, and H. Xue, "Logistic boosting regression for label distribution learning," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, pp. 4489–4497, July 2016.

[17] X. Yang, B.-B. Gao, C. Xing et al., "Deep Label Distribution Learning for Apparent Age Estimation," in *Proceedings of the 15th IEEE International Conference on Computer Vision Workshops, ICCVW 2015*, pp. 344–350, chl, December 2015.

[18] B.-B. Gao, C. Xing, C.-W. Xie, J. Wu, and X. Geng, "Deep label distribution learning with label ambiguity," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2825–2838, 2017.

[19] E. Elhamifar and R. Vidal, "Sparse subspace clustering: algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[20] H. Zhao, P. Zhu, P. Wang, and Q. Hu, "Hierarchical Feature Selection with Recursive Regularization," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 3483–3489, Melbourne, Australia, August 2017.

[21] X. Luo, X. Chang, and X. Ban, "Regression and classification using extreme learning machine based on L1-norm and L2-norm," *Neurocomputing*, vol. 174, pp. 179–186, 2016.

[22] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 793–804, 2014.

[23] B. Zhang, A. Perina, V. Murino, and A. Del Bue, "Sparse representation classification with manifold constraints transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 4557–4565, usa, June 2015.

[24] D. Luo, C. Ding, and H. Huang, "Towards structural sparsity: An explicit $\ell 2/\ell$ 0 approach," in *Proceedings of the 10th IEEE International Conference on Data Mining, ICDM 2010*, pp. 344–353, Australia, December 2010.

[25] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint l2,1-norms minimization," in *Advances in Neural Information Processing Systems*, pp. 1813–1821, MIT Press, 2010.

[26] S. Mosci, L. Rosasco, M. Santoro, A. Verri, and S. Villa, "Solving structured sparsity regularization with proximal methods," *Lecture Notes in Computer Science (including subseries Lecture Notes*

in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface, vol. 6322, no. 2, pp. 418–433, 2010.

[27] F. Wu, Y. Han, Q. Tian, and Y. Zhuang, "Multi-label boosting for image annotation by structural grouping sparsity," in *Proceedings of the 18th ACM International Conference on Multimedia ACM Multimedia 2010, (MM'10)*, pp. 15–24, ita, October 2010.

[28] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)*, pp. 333–342, ACM, Washington, DC, USA, July 2010.

[29] P. Zhu, W. Zuo, L. Zhang, Q. Hu, and S. C. K. Shiu, "Unsupervised feature selection by regularized self-representation," *Pattern Recognition*, vol. 48, no. 2, pp. 438–446, 2015.

[30] C. Hou, F. Nie, and D. Yi, "Feature selection via joint embedding learning and sparse regression," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, vol. 22, pp. 1324–1329, 2011.

[31] S. K. Shevade and S. S. Keerthi, "A simple and efficient algorithm for gene selection using sparse logistic regression," *Bioinformatics*, vol. 19, no. 17, pp. 2246–2253, 2003.

[32] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," in *Conference on Neural Information Processing Systems*, vol. 15, pp. 49–56, 2003.

[33] C.-N. Li, Y.-H. Shao, and N.-Y. Deng, "Robust L1-norm two-dimensional linear discriminant analysis," *Neural Networks*, vol. 65, pp. 92–104, 2015.

[34] W. J. McCalla, *Linear equation solution*, vol. 37, Springer, USA, 1988.

[35] S. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 2, pp. 300–307, 2007.

[36] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, New York, NY, USA, 2nd edition, 2001.

[37] F. Fahroo and I. M. Ross, "Direct trajectory optimization by a Chebyshev pseudospectral method," in *Proceedings of the American Control Conference*, vol. 6, pp. 3860–3864, 2000.

[38] K. X. Pearson, "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 50, no. 302, pp. 157–175, 1900.

[39] E. Deza and M.-M. Deza, *Dictionary of distances*, Elsevier, 2006.

[40] H.-T. Lin, C.-J. Lin, and R. C. Weng, "A note on Platt's probabilistic outputs for support vector machines," *Machine Learning*, vol. 68, no. 3, pp. 267–276, 2007.

[41] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research (JMLR)*, vol. 5, pp. 975–1005, 2004.

[42] R. Malouf, "A comparison of algorithms for maximum entropy parameter estimation," in *Proceedings of the proceeding of the 6th conference*, pp. 1–7, Not Known, August 2002.

[43] A. M. Chekroud, R. J. Zotti, Z. Shehzad et al., "Cross-trial prediction of treatment outcome in depression: A machine learning approach," *The Lancet Psychiatry*, vol. 3, no. 3, pp. 243–250, 2016.

[44] C. Xu, T. Liu, D. Tao, and C. Xu, "Local Rademacher complexity for multi-label learning," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1495–1507, 2016.