TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 01/11 pp135–145
DOI: 10.26599/TST.2019.9010052
Volume 26, Number 2, April 2021

LTSA-LE: A Local Tangent Space Alignment Label Enhancement Algorithm

Chao Tan*, Genlin Ji, Richen Liu, and Yanqiu Cao

Abstract: According to smoothness assumption, local topological structure can be shared between feature and label manifolds. This study proposes a new algorithm based on Local Tangent Space Alignment (LTSA) to implement the label enhancement process. In general, we first establish a learning model for feature extraction in label space and use a feature extraction method of LTSA to guide the reconstruction of label manifolds. Then, we establish an unconstrained optimization model based on the optimal theory presented in this paper. The model is suitable for solving problems with a large number of sample points. Finally, the experiment results show that the algorithm can effectively improve the training speed and multilabel dataset prediction accuracy.

Key words: smoothness assumption; feature manifold; label manifold; unconstrained optimization

1 Introduction

Multi-Label Learning (MLL) is a major topic in recent machine learning and pattern recognition studies. In an MLL framework, each instance is represented by a feature vector that can belong to multiple labels. MLL[1] deals with the case where an instance is associated with multiple labels, and its goal is to learn a multilabel predictor that maps an instance to a relevant label set. With the introduction of MLL, many scholars have conducted extensive research on this basis and proposed many effective algorithms.

This learning process works by mapping an instance and then assigning a label[2]. However, with the increase in the number of labels, the standard MLL methods that work in the original label space become impractical easily when training multilabel classifiers. For example, a large number of labels require a large amount time to train and test; thus, establishing an effective classification system is difficult. Usually, redundant information occurs in the label space, and the labels are generally related to each other. Therefore, some researchers have begun to study the method of dimensionality reduction in label space by using label correlations. The expectation is to improve classification accuracy and reduce training and prediction time for the entire model[3].

Some researchers have considered low-dimensional embedded label space and proposed many label space reduction methods. For example, Tai and Lin[4] attempted to reduce the amount of computation by seeking a major correlation between labels, especially for the datasets with numerous labels. Sun et al.[5] mapped the feature space and label space into the new space, where the correlation between the mapping of the two spaces is maximized. In these cases, the dimension of the label space is reduced to digest the information between the labels and learn more effectively.

However, some existing methods perform label space embedding without considering feature information, and a few methods can effectively utilize the local structure or label correlation of the feature space. Thus, these methods tend to lose some of the information, thereby seriously affecting the effectiveness of multilabel

• Chao Tan is with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, and also with the School of Computer Science and Technology, Nanjing Normal University, Nanjing 210023, China. E-mail: tutu_tanchao@163.com.
• Genlin Ji, Richen Liu, and Yanqiu Cao are with the School of Computer Science and Technology, Nanjing Normal University, Nanjing 210023, China. E-mail: glji@njnu.edu.cn; richen.liu@njnu.edu.cn; caoyanqiu1021@126.com.
* To whom correspondence should be addressed.
Manuscript received: 2019-06-25; revised: 2019-08-30; accepted: 2019-09-09

classification[3].

According to the aforementioned questions, we propose a new manifold-based label enhancement algorithm. First, we connect the feature and label spaces according to the smoothness assumption, and preserve the local geometry of the feature space by Local Tangent Space Alignment (LTSA). Then, reconstruction is conducted by a least squares programming problem from the feature manifold to the label space under the guidance of the feature information. The reconstruction can be achieved by a quadratic programming process. The mapping from the feature manifold to the label manifold is a regression process. The reconstruction process establishes an unconstrained optimization model based on Multi-output Support Vector Regression (MSVR)[6]. This method transforms the original optimization problem into an unconstrained optimization problem by introducing the maximum entropy function instead of a regularization term. Furthermore, the method avoids the difficulty of solving the MSVR constrained optimization problem in a large number of samples. However, the standard maximum entropy function method may lead to data overflow. In this paper, we improve the maximum entropy function and propose an MSVR model based on adaptive adjustment of Shannon entropy function, which guarantees convergence and uses quasi-Newton method. This algorithm is particularly suitable for problems with a large number of samples.

The main contributions of this paper are as follows:

• A learning model is established for feature extraction in label space by using a feature extraction method of LTSA to guide the reconstruction of label manifolds.

• Based on the MSVR constrained optimization model, the maximum entropy function is transformed into an unconstrained optimization problem instead of regularization term in the reconstruction process of least squares programming.

• An MSVR model is proposed on the basis of adaptive adjustment of the Shannon entropy function to improve the maximum entropy function.

This paper is organized as follows. First, the formulation of label enhancement and label enhancement based on manifold is discussed in Section 2. Then, the details of our algorithm are proposed in Section 3. Thereafter, the experiment results of the comparative study are reported in Section 4, and the conclusion is provided in Section 5.

2 Related Work on Label Enhancement

2.1 Formulation of label enhancement

First of all, the main notations used in this paper are listed as follows: the instance variable is denoted by $\boldsymbol{x}$, the particular i-th instance is denoted by $\boldsymbol{x}_i$, the label variable is denoted by y, and the particular j-th label is denoted by y_j. The logical label vector of x_i is denoted by $\boldsymbol{L}_i = (l_i^1, l_i^2, \ldots, l_i^c)$, where $\boldsymbol{L}_i \in \{0,1\}^c$ and c is the number of possible labels. The description degree of y to $\boldsymbol{x}$ is denoted by d_x^y, where $d_x^y \in [0,1]$ and $\sum_y d_x^y = 1$. The label distribution of $\boldsymbol{x}_i$ is denoted by $\boldsymbol{D}_i = (d_i^1, d_i^2, \ldots, d_i^c)$, where $\boldsymbol{D}_i \in [0,1]^c$.

Then, the label distribution learning for label enhancement can be defined as follows: given a training set $S = \{(\boldsymbol{x}_i, \boldsymbol{L}_i) | 1 \leqslant i \leqslant n\}$, the goal of label enhancement is to transform the logical label vector $\boldsymbol{L}_i$ of $\boldsymbol{x}_i$ to the label distribution $\boldsymbol{D}_i$ according to the correlation between labels contained in S, thereby obtaining a Label Distribution Learning (LDL) training set $\varepsilon = \{(\boldsymbol{x}_i, \boldsymbol{D}_i) | 1 \leqslant i \leqslant n\}$[7].

2.2 Label enhancement based on manifold

The label enhancement algorithm based on manifold[8] assumes that the data are distributed in the feature and label manifolds. This algorithm connects the manifolds of the two spaces according to the smoothness assumption, so that the label manifold is reconstructed with the topological structure from the feature manifold and the logical label of this instance is enhanced to the label distribution on this basis.

As many graph-based learning methods do, the topological structure from the feature space of multiple label training set S can be represented by a graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \hat{\boldsymbol{W}})$, where $\boldsymbol{V}$ is the vertex set, $\boldsymbol{E}$ is the edge set, and $\hat{\boldsymbol{W}} = (\hat{w}_{ij})_{n\times n}$ is the weight matrix with the edge. First, we assume that the manifold of the instance distribution satisfies local linearity, that is, each instance $\boldsymbol{x}_i$ can be optimally reconstructed using a linear combination of its k-nearest neighbors. The reconstructed weight matrix is to induce the minimization of

$$\Omega(\hat{\boldsymbol{W}}) = \sum_{i=1}^{n} \| \boldsymbol{x}_i - \sum_{i \neq j} \hat{w}_{ij} \boldsymbol{x}_j \|^2 \tag{1}$$

where $\hat{w}_{ij} = 0$ unless $\boldsymbol{x}_j$ is one of $\boldsymbol{x}_i$'s k-nearest neighbors. The constraint is $\mathbf{1}^{\mathrm{T}} \hat{\boldsymbol{W}}_i^{\mathrm{T}} = 1$ for translation invariance, where $\mathbf{1}^{\mathrm{T}}$ represents the vector consisting of all 1 and $\hat{\boldsymbol{w}}_i$ is the i-th row of $\hat{\boldsymbol{W}}$. According to the smoothness assumption[9], the instances with similar

features are likely to have similar labels. Thus, one can migrate the topological structure of the feature space into the label space, that is, the feature and the label space share the same local linear reconstruction weight matrix $\hat{\boldsymbol{W}}$. Therefore, the label distribution of the label space can infer to the minimization of

$$\Phi(\boldsymbol{D}) = \sum_{i=1}^{n} \|\boldsymbol{D}_i - \sum_{i \neq j} \hat{w}_{ij} \boldsymbol{D}_j\|^2 \tag{2}$$

where $\boldsymbol{D}_i = (d_i^1, d_i^2, \ldots, d_i^c)$ denotes the distribution of $\boldsymbol{x}_i$. A constraint is added on d_i to introduce logical label $\boldsymbol{L}_i = (l_i^1, l_i^2, \ldots, l_i^c)$,

$$\forall 1 \leqslant i \leqslant n,\ 1 \leqslant l_i \leqslant c,\ l_i^c d_i^c \geqslant \lambda (\lambda > 0) \tag{3}$$

To facilitate the construction of the aforementioned constraints, the logical label vector defined in Ref. [8] is $\boldsymbol{L}_i \in \{-1, 1\}^c$, instead of $\boldsymbol{L}_i \in \{0, 1\}^c$, which is commonly used in other methods. However, no difference exists between these two vectors. After solving the preceding quadratic programming problem, we can obtain the label distribution $\boldsymbol{D}_i$ by normalization, and thereafter obtain the label distribution training set $\varepsilon = \{(\boldsymbol{x}_i, \boldsymbol{D}_i) | 1 \leqslant i \leqslant n\}$.

The manifold-based method reconstructs the feature and space manifolds according to the smoothness assumption that migrates the topological relationship of the feature space into the label space. Then, the method establishes the relationship between the correlation between the instances and the correlation between the labels, thereby establishing the logic. Finally, the logical labels are enhanced to label distribution[1].

2.3 Label enhancement for label distribution learning

To solve the Label Enhancement (LE) problem, Xu et al.[10] introduced an existing algorithm that can be used for LE and proposed a new method of label enhancement called label distribution learning. The label distribution is recovered from the logical labels in the training set by utilizing the topology information of the feature space and the correlation between the labels.

2.4 Label embedding based on multi-scale locality preservation

Peng et al.[11] proposed a new label distribution learning algorithm by using local sample correlation. Label distribution is learned by leveraging sample correlation locally.

2.5 Multi-label learning with label enhancement

Shao et al.[12] proposed an effective MLL method called label-enhanced MLL, which is based on label enhancement. Through this approach, problems were developed by combining numerical label and label-enhanced regression into a unified framework, in which numerical labels and predictive models are learned jointly.

3 Proposed Algorithm

3.1 Preprocessing on training set: Establish the correlation between manifolds and label spaces

In this section, we explore multilabel manifold learning. To study the label manifold, we have to extend the label space to the Euclidean space, because the traditional label space is logical and the label vector is a logical label. Here we extend the label vector from logic to real numbers called numerical labels. Hou et al.[8] mentioned that numerical labels carry more semantic information and can describe instances more comprehensively than logical labels. To reconstruct the label manifold, we first preprocess the training set, and then establish the association between the manifold and label spaces. Finally, we extend the feature space to the label space and obtain the numerical label.

Inspired by the LTSA algorithm proposed by Zhang and Zha[13], we approximate the feature manifold by overlapping local linear neighborhoods and obtain the topological structure of the feature space by using LTSA.

To obtain the mapping coordinates of the high-dimensional data set $\boldsymbol{X}$ in the low-dimensional space, we use the LTSA algorithm to determine the global coordinates by applying LTSA. The dataset is denoted by $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$ with N high-dimensional coordinates.

We minimize

$$\min_{x, L, \Theta} \|\boldsymbol{X}_i - (\bar{\boldsymbol{x}}_{\mathrm{i}} \boldsymbol{e}^{\mathrm{T}} + \boldsymbol{L}\boldsymbol{\Theta})\|^2 \tag{4}$$

where $\boldsymbol{X}_i = [\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{ik}]$ is a matrix consisting of its k-nearest neighbors including $\boldsymbol{x}_i$, in terms of the Euclidean distance, $\bar{\boldsymbol{x}}_{\mathrm{i}}$ is the average of $\boldsymbol{X}_i$, $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k]$ is the weight matrix, $\boldsymbol{L}$ is the optimal alignment matrix consisting of d-dimensional orthogonal column vectors $\boldsymbol{L}_i$, given by the eigenvectors of $\boldsymbol{X}_i(\boldsymbol{I} - \boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}/k)$ corresponding to d eigenvalues in descending order, $\boldsymbol{e}$ is the k-dimensional column vector whose element are all ones, and $\boldsymbol{I}$ is the k-dimensional identity matrix. Thus,

$$\boldsymbol{X}_i = \bar{\boldsymbol{x}}_{\mathrm{i}} + \boldsymbol{L}_i \boldsymbol{\theta}_i + e_i \tag{5}$$

where $e_i = (\boldsymbol{I} - \boldsymbol{L}_i \boldsymbol{L}_i^{\mathrm{T}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_{\mathrm{i}})$ is the reconstruction error.

The objective of this study is to recover the low-dimensional coordinate $\boldsymbol{Y}_i$ from the least squares and reconstruct the topological structure of the label manifold from the feature manifold and the existing logical labels. Furthermore, the study aims to determine that a reasonable label distribution d_i can be generated for the instance x_i.

Our model is described as follows:

$$\boldsymbol{Y}_i = \frac{1}{k}\boldsymbol{Y}_i\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}} + \boldsymbol{L}_i\boldsymbol{\Theta}_i^{\mathrm{T}} + \boldsymbol{E}_i \tag{6}$$

where $\boldsymbol{E}_i = [e_{i1}, \ldots, e_{ik}]$ is the local reconstruction error matrix. To minimize the reconstruction error, we utilize quadratic programming on the model to obtain the following optimization function,

$$\min_{1\leqslant i\leqslant N}\sum_{i=1}^{N}\left(\|\boldsymbol{E}_i\|_{\mathrm{F}}^2 = \left\|\boldsymbol{Y}_i\left(\boldsymbol{I}_k - \frac{1}{k}\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right) - \boldsymbol{L}_i\boldsymbol{\Theta}_i^{\mathrm{T}}\right\|_{\mathrm{F}}^2\right) \tag{7}$$

To obtain a unique solution, we express the reconstruction error as

$$\sum_{i=1}^{N}\left\|\boldsymbol{Y}_i\left(\boldsymbol{I}_k - \frac{1}{k}\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)\left(\boldsymbol{I}_k - \boldsymbol{\Theta}_i\boldsymbol{\Theta}_i^{\dagger}\right)\right\|_{\mathrm{F}}^2 = \mathrm{trace}\left(\boldsymbol{T}\boldsymbol{\Phi}\boldsymbol{T}^{\mathrm{T}}\right) \tag{8}$$

where $\boldsymbol{\Theta}_i^{\dagger}$ is the Moor-Penrose generalized inverse of $\boldsymbol{\Theta}_i$, $\boldsymbol{T}$ is a matrix composed by the low-dimensional embeddings, $\boldsymbol{\Phi} = \sum_{i=1}^{N}\boldsymbol{S}_i\boldsymbol{H}_i\boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{S}_i^{\mathrm{T}}$, $\boldsymbol{S}_i$ is the 0-1 selection matrix, such that $\boldsymbol{T}\boldsymbol{S}_i = \boldsymbol{T}_i$, and $\boldsymbol{H}_i$ is given by

$$\boldsymbol{H}_i = \left(\boldsymbol{I}_k - \frac{1}{k}\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)\left(\boldsymbol{I}_k - \boldsymbol{Q}_i\boldsymbol{Q}_i^{\dagger}\right) \tag{9}$$

The low-dimensional embeddings that minimize the reconstruction error can be solved by the eigenvectors corresponding to the d largest eigenvalues of the matrix $\boldsymbol{\Phi}$.

3.2 Unconstrained optimization model

The feature manifold is represented by a graph and approximated by overlapping local linear neighborhood patches. The edge weights in each patch can be calculated by a least squares programming method. The label manifold and transferred local topological reconstructed structure can be reconstructed based on the feature manifold and existing logical labels. The reconstruction can be accomplished through a quadratic method.

As shown in the previous section, in the manifold space, the feature manifold is represented by a graph and approximated by overlapping local linear neighborhoods. An optimization model that can solve the edge weight in each patch is expressed as Eq. (6).

We propose a new LE algorithm called LTSA Label Enhancement (LTSA-LE) in this paper. Given a training set, we construct a feature matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]$ and a logical label matrix $\boldsymbol{L} = [l_1, l_2, \ldots, l_N]$. Our goal is to recover the label distribution matrix $\boldsymbol{D} = [d_1, d_2, \ldots, d_N] \in \mathbf{R}^{c\times N}$ from the logical label matrix $\boldsymbol{L}$. To solve this problem, we consider the model:

$$\boldsymbol{D}_i = \boldsymbol{W}\boldsymbol{\theta}_i + \boldsymbol{r} \tag{10}$$

where $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_d] \in \mathbf{R}^{c\times d}$, $\boldsymbol{\theta}_i \in \mathbf{R}^d$ is the nonlinear transformation of the sample point in the low-dimensional space $\mathbf{R}^d$ and is the optimal solution of Eq. (6), $\boldsymbol{r} \in \mathbf{R}^c$ is the model parameter.

As the information in the label distribution is inherited from the initial logical label, we choose the least squares loss function, which is actually a multiple output regression problem in the case of multiple labels. After determining d_i by solving the quadratic programming problem, the label distribution is obtained by normalization,

$$F(\boldsymbol{W}, \boldsymbol{r}) = \frac{1}{2}\sum_{j=1}^{c}\|\boldsymbol{w}_j\|^2 + C\sum_{i=1}^{N}\mathrm{Loss}(u_i) \tag{11}$$

where Loss( ) is the loss function and $\boldsymbol{u}_i = \boldsymbol{y}_i - \boldsymbol{W}\boldsymbol{\theta}_i - \boldsymbol{r}$, C is the model parameter.

Then, $\boldsymbol{y}_i \in \{-1, +1\}^c$ represents a numeric label vector, where we use -1 instead of 0 to indicate that it is independent of the instance. This study explores manifolds in label space and treats labels as numbers. The label set contains additional semantic information, which is beneficial for the learning process.

The multilabel manifold learning algorithm proposed by Hou et al.[8] solves a quadratic programming problem with constraints and obtains the label distribution through optimization. Constraints are utilized to express constraints on problems, such as structural features in the real dataset, local neighborhood relationships, and manifold structure information. However, the solution to the quadratic programming problem with constraints will face limitations in time and memory when the number of samples is large.

Based on the MSVR constrained optimization model, we establish an unconstrained optimization model based on the optimal theory. By introducing the maximum entropy function, we transform the original optimization problem into an unconstrained optimization problem, which solves the difficulty that the MSVR constrained optimization problem seeks the solution from a large number of samples. However, the standard maximum entropy function method may lead to data overflow. In this study, we improve the maximum entropy function and propose an MSVR model based on adaptively adjusting the Shannon entropy function, which ensures

convergence and utilizes the quasi-Newton method to obtain the model. In particular, the model is suitable for problems with a large number of sample points.

3.3 Multi-output regression adaptive weighting strategy based on Shannon entropy

The existing multi-output regression algorithm almost defaults to the same premise, that is, the weight parameter of each sample contributes equally to the final classification result. In fact, some sample spaces often have serious spatial overlaps and are inseparable. The suitable treatment should give different degrees of importance according to the divisible characteristics of the sample, so that the classification results can be optimal. However, artificially setting importance weights is unreasonable. Thus, using the Shannon entropy theory[14], we propose an adaptive weighted term with multi-output regression technique, and then introduce the viewing angle weight coefficient w_k. Finally, we reconstructed objective function formula according to the condition $\sum_{k=1}^{N} \tilde{w}_k = 1$ and $\tilde{w}_k \geqslant 0$.

We consider weights as probability distributions expressed with Shannon entropy as

$$f(\tilde{w}) = \sum_{j=1}^{c} \tilde{w}_j \ln \tilde{w}_j \tag{12}$$

According to the definition of the unconstrained optimization model in Section 3.2 and the multi-output regression adaptive weighting strategy, the objective function in this section is reconstructed as follows:

$$\Gamma(\boldsymbol{W}, \boldsymbol{e}) = F(\boldsymbol{W}, \boldsymbol{e}) + f(\tilde{w}) \tag{13}$$

We define the adaptive adjustment Shannon entropy function as an adaptive weighting part based on Shannon entropy theory,

$$\Gamma(\boldsymbol{W}, \boldsymbol{e}) = \frac{1}{2} \sum_{j=1}^{c} \|\tilde{w}_j\|^2 + C \sum_{i=1}^{N} \mathrm{Loss}(u_i) + \sum_{j=1}^{c} \alpha \tilde{w}_j \ln \tilde{w}_j \tag{14}$$

where α is an adaptive adjustment factor. We use an iterative quasi-Newton method called IRWLS[6,8] to optimize Eq. (14). In terms of convergence, we know that the Newton algorithm is convergent; thus, the entire algorithm has convergence.

The main procedure of the LTSA-LE algorithm is presented in Algorithm 1.

4 Experiment

4.1 Experimental setup

4.1.1 Comparing algorithms

We select two well-established MLL algorithms to compare with the performance of LTSA-LE: an MLL algorithm based on a neural network model (called BP-MLL)[15] and multilabel manifold learning (called ML^2)[8]. We also selected three types of LDL algorithms: Label Distribution Support Vector Regressor (LDSVR)[16], Conditional Probability Neural Network (CPNN)[17], and Algorithm Adaptation k-Nearest Neighbors (AA-KNN)[7].

Algorithm 1 Procedure of LTSA-LE

Input: A training sample matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbf{R}^{d \times N}$, a numeric label vector $\boldsymbol{y}_i \in \{-1, +1\}^c$, where -1 indicates it is independent of the instance x_i.

Output: Label distribution $\boldsymbol{D}$ for the multilabel sample set $\boldsymbol{X}$.

1: $i \leftarrow 1$, $j \leftarrow 1$.
2: Compute the optimal solution of Eq. (6), and obtain $\boldsymbol{\theta}_i \in \mathbf{R}^d$, which is the nonlinear transformation of the sample point in the low-dimensional space $\mathbf{R}^d$.
3: **repeat**
4: Optimize $\Gamma(\boldsymbol{W}, \boldsymbol{e})$ according to Eq. (14).
5: Update $f(\tilde{w})$ according to Eq. (12).
6: Update Eq. (14) via the Iterative Re-Weighted Least Square (IRWLS) procedure.
7: $j \leftarrow j + 1$, $i \leftarrow i + 1$.
8: **until** convergence is reached.
9: Return $\boldsymbol{D}$ according to Eq. (10).

Table 1 Characteristic of multilabel datasets.

Dataset	S	T	dim(S)	$L(S)$	LCard(S)	LDen(S)	DL(S)	$F(S)$
Emotions	415	178	72	6	1.869	0.311	27	numeric
Medical	645	333	1449	45	1.245	0.028	94	nominal
Cal500	250	252	68	174	26.044	0.150	502	numeric
Birds	320	325	260	19	1.014	0.053	133	numeric
Enron	1123	579	1001	53	3.378	0.064	753	nominal
Yeast	1200	1217	103	14	4.237	0.303	198	numeric
Image	1000	1000	294	5	1.236	0.247	20	numeric
Scene	1211	1196	294	6	1.074	0.179	15	numeric
Corel5k	2500	2500	499	374	3.522	0.009	3175	nominal
Bibtex	3700	3695	1836	159	2.402	0.015	2856	nominal

4.1.2 Datasets

To compare our algorithm with the state-of-the-art MLL algorithms, we selected 10 real-world multilabel datasets for performance evaluation. Table 1 summarizes the detailed features of these datasets selected from the Mulan website (http://mulan.sourceforge.net/datasets-mlc.html, 2019-03-01) and sorted in ascending order. S is the number of examples, T is the number of testing samples, $\dim(S)$ denotes the feature dimensions, $L(S)$ represents the number of class labels, and $\mathrm{LCard}(S)$ is the label cardinality. Other multilabel statistics include the label density $\mathrm{LDen}(S)$, distinct label sets $\mathrm{DL}(S)$, and feature type $F(S)$. Half of them are regular-sized and half are large-scale; thus, the dataset covered a wide range of multilabel attributes.

4.1.3 Evaluation metrics

When comparing LTSA-LE to traditional MLL methods, we selected five widely used evaluation metrics for multilabel learning: Hamming loss, one error, coverage, ranking loss, and average precision[18]. For average precision, the larger the value, the better the performance. For the other four metrics, the smaller the values, the better the performance.

When comparing LTSA-LE with LDL algorithms[7, 16, 17], we used six representative label distribution evaluation indicators[7]: Chebyshev distance (Cheb), Clark distance (Clark), Canberra distance (Canber), Kullback-Leibler divergence (KL-div), Cosine coefficient (Cosine), and intersectional similarity (Intersec). The first four are distance measures and the last two are similarity measures.

4.2 Experiment results on MLL

The first part of the experiment compares LTSA-LE with traditional MLL algorithms on 10 real-world multilabel datasets. Half the examples on each dataset were selected randomly as a training set, and the remaining half were used to form a test set. We used 10-fold cross-validation on each dataset, set $C = 0.5$ and $\alpha = 0.5$. We recorded the average performance of each algorithm on five MLL evaluation metrics in Tables 2, 3, 4, 5, and 6. Table 7 shows the time complexity comparison. The $\downarrow$ after the evaluation index means the smaller the better, and the $\uparrow$ means the larger the better. Bold font indicates the best performance among the algorithms.

Table 2 Comparison of MLL and multilabel distribution algorithms on Hamming loss ↓.

Algorithm	Dataset										Average rank
	Yeast	Emotions	Medical	Cal500	Birds	Image	Scene	Enron	Corel5k	Bibtex	
BP-MLL	0.4500	0.2987	0.0290	0.1472	0.0683	0.3056	0.2904	0.0682	**0.0094**	0.0160	3.000
ML^2	0.2267	**0.2865**	0.3405	0.3701	0.1330	0.2450	0.1803	0.2049	0.3922	0.0747	3.600
LDSVR	0.3037	0.2996	0.9721	0.1488	**0.0517**	0.7516	0.1810	0.0677	0.9907	0.0149	3.450
CPNN	0.6964	0.7097	0.9732	0.8522	0.9491	0.7522	0.8194	0.9339	0.9907	0.9853	5.950
AA-KNN	0.2297	0.3006	0.0184	0.1814	0.0748	0.2158	0.1134	0.0705	0.0114	0.0165	3.100
LTSA-LE	**0.1958**	0.3024	**0.0145**	**0.1400**	0.1705	**0.1656**	**0.0962**	**0.0558**	0.0096	**0.0127**	1.900

Table 3 Comparison of MLL and multilabel distribution algorithms on ranking loss ↓.

Algorithm	Dataset										Average rank
	Yeast	Emotions	Medical	Cal500	Birds	Image	Scene	Enron	Corel5k	Bibtex	
BP-MLL	0.4450	0.4803	0.2445	**0.1996**	0.3964	0.7956	0.5992	0.3738	**0.2695**	0.4764	2.800
ML^2	0.3325	0.3579	0.4984	0.4981	0.4051	0.2470	0.1915	0.4493	0.5101	0.2199	2.900
LDSVR	0.4974	0.5899	0.5000	0.5005	0.4374	0.5000	0.6556	0.4741	0.5000	0.5012	4.300
CPNN	0.9708	0.8511	0.8982	0.8621	**0.3132**	0.8892	0.8609	0.9621	0.4990	0.6954	5.100
AA-KNN	0.5054	0.4283	0.5039	0.7750	0.7335	0.3139	0.1838	0.8563	0.9444	0.7416	4.600
LTSA-LE	**0.3244**	**0.2426**	**0.1453**	0.4648	0.3158	**0.1406**	**0.0713**	**0.3509**	0.4393	**0.1052**	1.300

Table 4 Comparison of MLL and multilabel distribution algorithms on one error ↓.

Algorithm	Dataset										Average rank
	Yeast	Emotions	Medical	Cal500	Birds	Image	Scene	Enron	Corel5k	Bibtex	
BP-MLL	0.7034	0.7022	0.4024	0.1071	0.7989	0.6710	0.8269	0.2642	0.9716	0.4547	4.300
ML^2	0.4286	0.6667	0.4737	**0.0827**	0.9474	0.2000	0.6667	0.8846	0.9564	0.6792	4.000
LDSVR	0.4286	0.6667	0.5000	0.8563	0.4990	0.5000	0.4999	0.9615	0.4890	0.9497	4.500
CPNN	0.0714	**0.3333**	0.4290	0.3333	0.8421	0.5470	0.3333	0.5050	0.4400	0.9874	3.400
AA-KNN	0.4999	0.4899	**0.1579**	0.5862	0.4737	0.4990	0.5000	0.4808	0.4419	0.7688	3.100
LTSA-LE	**0.0000**	0.5000	0.3947	0.7816	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	1.700

Table 5 Comparison of MLL and multilabel distribution algorithms on coverage ↓.

Algorithm	Dataset										Average rank
	Yeast	Emotions	Medical	Cal500	Birds	Image	Scene	Enron	Corel5k	Bibtex	
BP-MLL	0.8990	0.3089	0.2955	1.3386	0.4415	2.1460	2.0761	**0.2369**	0.1980	0.7356	5.100
ML^2	0.8950	0.1723	0.7684	0.2313	0.3031	0.9962	1.0617	0.5029	0.1912	0.3513	4.200
LDSVR	0.8982	**0.1568**	0.2087	0.2284	0.3014	0.9608	1.0843	0.4936	1.5023	0.3382	3.250
CPNN	**0.8845**	0.1703	0.2081	0.2316	**0.2309**	0.9648	1.0773	0.5028	1.5023	0.3598	3.650
AA-KNN	0.8794	0.1661	0.1374	0.2315	0.2899	0.9644	1.0505	0.4956	1.5126	0.3585	3.300
LTSA-LE	0.8846	0.1586	**0.0598**	**0.2275**	0.2523	**0.9538**	**0.1048**	0.4456	**0.1502**	**0.2596**	1.500

Table 6 Comparison of MLL and multilabel distribution algorithms on average precision ↑.

Algorithm	Dataset										Average rank
	Yeast	Emotions	Medical	Cal500	Birds	Image	Scene	Enron	Corel5k	Bibtex	
BP-MLL	0.4297	0.5161	0.2081	**0.4783**	**0.2460**	0.5111	0.4200	**0.2057**	0.2012	0.0659	2.600
ML^2	0.4366	0.4442	0.4683	0.1644	0.1407	0.4905	0.3078	0.1234	**0.2930**	**0.3872**	3.200
LDSVR	0.3965	0.4900	0.0480	0.1676	0.0759	0.2729	0.7859	0.0747	0.0141	0.0226	4.750
CPNN	0.3064	0.3123	0.0467	0.1598	0.1013	0.2645	0.2954	0.0828	0.0141	0.0182	5.750
AA-KNN	0.4779	0.4926	0.3692	0.1705	0.1131	0.5954	0.7649	0.1201	0.0252	0.1111	3.100
LTSA-LE	**0.4825**	**0.5567**	**0.5631**	0.1829	0.1602	**0.7105**	**0.7957**	0.1843	0.0278	0.3678	1.600

Table 7 Comparison of MLL and multilabel distribution algorithms on time (s) ↓.

Algorithm	Dataset										Average rank
	Yeast	Emotions	Medical	Cal500	Birds	Image	Scene	Enron	Corel5k	Bibtex	
BP-MLL	4.6563	0.7020	6.7080	10.2337	1.5625	1.5288	1.6380	13.2757	218.3234	122.6948	5.600
ML^2	0.4836	0.0005	0.1716	0.1404	0.0469	0.2500	0.3438	0.1875	1.5444	4.5625	3.300
LDSVR	0.1406	**0.0001**	0.1872	0.0780	**0.0001**	0.2496	0.2652	0.1716	1.0156	3.5781	2.300
CPNN	0.1092	**0.0001**	0.5616	0.2496	0.0468	0.1404	0.0780	0.9048	15.0385	25.6466	3.100
AA-KNN	3.9936	0.2340	9.8593	0.2184	0.5460	3.9624	5.7876	14.7109	21.4345	107.28	5.300
LTSA-LE	**0.0938**	**0.0001**	**0.0781**	**0.0469**	0.1404	**0.0938**	**0.0624**	**0.1406**	**0.7969**	**2.9531**	1.400

The experimental results show that on the regular-sized and large-scale datasets, LTSA-LE ranks first in more than half of the evaluation metrics. Particularly on large-scale datasets, these metrics fully validate the effectiveness of LTSA-LE for MLL.

4.3 Time performance comparison

As Table 7 shows, our algorithm LTSA-LE performs effectively among the well-established MLL algorithms, especially on large-scale datasets. Although the accuracy of some algorithms (such as BP-MLL, ML^2, and AA-KNN) have improved, their time complexity increases rapidly as the sample set size increases. Combined with the recognition accuracy experiments in the previous section, LTSA-LE maintains a good average performance and is remarkably adept in the label manifold learning of a large-scale dataset.

4.4 Friedman test

The Friedman test[19] is a nonparametric equivalent of repeated measurement ANOVA. The test separately ranks the algorithm for different datasets, and the algorithms with best performances get the first rank, the second rank, and so on, as shown in the average rank in Tables 2, 3, 4, 5, 6, and 7.

Let r_i^j represents the rank of the j-th of the k algorithms on the i-th of the N data sets. The Friedman test compares the average rank of the algorithm, i.e., $R_j = \sum_i r_i^j / N$. Under the null hypothesis, this condition means that all the algorithms are equivalent; thus, their ranks R_j should be equal. The Friedman statistic is calculated as follows:

$$\chi_{\mathrm{F}}^2 = \frac{12N}{k\,(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{15}$$

which obeys the χ_{F}^2 distribution with a freedom degree of $k-1$, where N and k are large enough (generally $N \geqslant 10$ and $k > 5$). The exact threshold for a small number of algorithms and data sets has been calculated.

Demsar[19] showed that Friedman estimate of χ_{F}^2 is conservative and suggested a better statistic that satisfies the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees

of freedom as follows:

$$F_{\mathrm{F}} = \frac{(N-1)\,\chi_{\mathrm{F}}^2}{N\,(k-1)-\chi_{\mathrm{F}}^2} \tag{16}$$

If the null hypothesis is rejected, then we can proceed with a post-hoc test. The Bonferroni-Dunn test is used in this study. The test method controls the overall error level by dividing α, to one of the original $k-1$ points. Thus, we only need to test whether the new algorithm is better than the existing algorithm. We do not need a two-two pairwise comparison, and we only need to set the new algorithm as the control algorithm. Whenever the performance between an existing algorithm i and the new algorithm j is significantly different, we only need to compare whether the difference between their corresponding average ranks (i.e., $|R_j - R_i|$) is greater than the significant difference. If greater, the algorithms are considered different; otherwise, they are considered similar.

Table 8 denotes the Friedman statistics F_{F} and the corresponding critical value on each evaluation metric.

The significance level CD is defined as follows:

$$\mathrm{CD} = q_\alpha \sqrt{\frac{k\,(k+1)}{6N}} \tag{17}$$

where q_α is a critical value for post-hoc tests after the Friedman test.

Owing to the existence of 6 algorithms and 10 datasets, F_{F} obeys the F-distribution with degrees of freedom of $6-1=5$ and $(6-1)\times(10-1)=45$. $F(5,45)=2.42$ at $\alpha=0.05$. Thus, we reject the original null hypothesis, i.e., the six algorithms are considered to be significantly different.

Then, we use the Bonferroni-Dunn test with LTSA-LE as the control algorithm to test whether other algorithms are significantly different from LTSA-LE. From Table 9 we can find that when six algorithms exist, $q_{0.05} = 2.576$; thus $\mathrm{CD} = 2.576\sqrt{\dfrac{6\times(6+1)}{6\times 10}} = 2.155$.

In Table 2, the difference between the average ranks between CPNN and LTSA-LE is $5.950 - 1.900 = 4.050 > 2.155$. Thus, we think that CPNN and LTSA-LE are significantly different. The difference between

Table 8 Friedman statistics F_{F} in terms of each evaluation metric, the critical value is 2.42 when the significance level is 0.05 (the number of comparing algorithms, k, is 6 and the number of datasets, N, is 10).

Evaluation metric	F_{F}	Evaluation metric	F_{F}
Hamming loss	9.50	Coverage	6.25
Ranking loss	12.29	Average precision	16.38%
One error	3.91	Time	2.75 s

Table 9 Critical values for post-hoc tests after the Friedman test.

Number of classifiers	$q_{0.05}$	$q_{0.10}$	Number of classifiers	$q_{0.05}$	$q_{0.10}$
2	1.960	1.645	7	2.638	2.394
3	2.241	1.960	8	2.690	2.450
4	2.394	2.128	9	2.724	2.498
5	2.498	2.241	10	2.773	2.539
6	2.576	2.326			

the average ranks of the other four algorithms (BP-MLL, ML^2, LDSVR, and AA-KNN) and LTSA-LE < 2.155, which is not significantly different. Thus, we can approximate an improvement of these algorithms from LTSA-LE.

4.5 Experimental results on LDL

We conduct quantitative analysis on the performance of four LDL algorithms on the datasets in Table 10. Tables 10 – 15 provide the comparison results of various types of LDL algorithms on six evaluation metrics: Cheb, Clark, Canber, KL-div, Cosine, and Intersec, respectively. As described in Section 4.1, we calculate the average ranks of the corresponding algorithms based on the six metrics in the last row of each table (bold font indicates the best performance on each dataset). We obtain the rankings of the algorithms on six measures according to their average ranks, LTSA-LE $\succ$ CPNN $\approx$ LDSVR $\succ$ AA-KNN.

As the results show, LTSA-LE performs best on all six measures. The reason might be that applying the kernel technique allows LTSA-LE to solve the problem of higher dimensionality and therefore obtain a

Table 10 Experimental results on the real-world datasets measured by the Cheb ↓.

Dataset	LDSVR	CPNN	AA-KNN	LTSA-LE
Yeast-alpha	0.0080	**0.0073**	0.0090	**0.0073**
Yeast-cdc	0.0141	0.0138	0.0229	**0.0083**
Yeast-elu	0.0267	0.0262	0.0189	**0.0078**
Yeast-diau	0.0414	0.0505	0.0572	**0.0194**
Yeast-heat	0.0481	0.0504	0.0669	**0.0118**
Yeast-spo	0.1774	0.1969	0.1830	**0.0268**
Yeast-cold	0.0773	0.0727	0.0833	**0.0172**
Yeast-dtt	0.0417	0.0436	0.0543	**0.0074**
Yeast-spo5	0.1722	0.1751	0.2006	**0.0491**
Yeast-spoem	0.3369	0.2869	0.1512	**0.0125**
Human Gene	0.0179	0.0202	0.0140	**0.0130**
Natural Scene	0.4045	0.1877	0.2473	**0.1291**
Movie	0.1697	0.1420	0.1544	**0.1359**
s-JAFFE	0.1300	0.1416	0.1183	**0.0576**
s-BU_3DFE	0.1790	0.1745	0.2229	**0.0794**
Average rank	2.97	2.77	3.23	1.03

Table 11 Experimental results on real-world datasets measured by the Clark ↓.

Dataset	LDSVR	CPNN	AA-KNN	LTSA-LE
Yeast-alpha	0.1713	0.1141	0.1262	**0.1021**
Yeast-cdc	0.2178	0.2005	0.2857	**0.1186**
Yeast-elu	0.2494	0.2351	0.2386	**0.1024**
Yeast-diau	0.2890	0.3287	0.3393	**0.0927**
Yeast-heat	0.2534	0.2492	0.3438	**0.0545**
Yeast-spo	0.4690	0.5093	0.4928	**0.1204**
Yeast-cold	0.1979	0.2077	0.2392	**0.0452**
Yeast-dtt	0.1244	0.1366	0.1363	**0.0181**
Yeast-spo5	0.4098	0.4233	0.4501	**0.0948**
Yeast-spoem	0.5671	0.5117	0.3180	**0.0176**
Human Gene	1.0551	1.0739	1.3913	**0.9660**
Natural Scene	2.3735	2.1469	2.1008	**2.0813**
Movie	0.7311	0.6558	0.5541	**0.4177**
s-JAFFE	0.3860	0.5163	0.3324	**0.2401**
s-BU_3DFE	0.4379	0.4457	0.5710	**0.2573**
Average rank	2.87	2.93	3.20	1.00

Table 12 Experimental results on real-world datasets measured by the Canber ↓.

Dataset	LDSVR	CPNN	AA-KNN	LTSA-LE
Yeast-alpha	0.5686	0.3792	0.4057	**0.3315**
Yeast-cdc	0.7528	0.6780	0.9368	**0.3230**
Yeast-elu	0.7263	0.7003	0.6932	**0.3064**
Yeast-diau	0.6489	0.7181	0.7262	**0.1648**
Yeast-heat	0.5499	0.5282	0.7659	**0.1114**
Yeast-spo	0.9634	1.0667	0.9917	**0.2340**
Yeast-cold	0.3574	0.3990	0.4401	**0.0723**
Yeast-dtt	0.2436	0.2709	0.2184	**0.0296**
Yeast-spo5	0.5826	0.5843	0.6554	**0.1505**
Yeast-spoem	0.7600	0.6749	0.3972	**0.0249**
Human Gene	6.2419	**6.1261**	9.6774	6.1365
Natural Scene	6.8702	5.5629	**5.2013**	5.2951
Movie	1.2497	1.2530	0.9741	**0.7652**
s-JAFFE	0.7755	1.0884	0.6243	**0.5023**
s-BU_3DFE	0.9382	0.9765	1.1770	**0.5483**
Average rank	2.93	2.93	3	1.13

more discriminative feature space without compromising computational feasibility. CPNN is based on the multilayer neural network. LTSA-LE appears steadier than CPNN with the decrease of training data mainly because CPNN learns the model from the training data. Consequently, LTSA-LE relies less on the training data than CPNN does. AA-KNN is approximate to LDSVR because AA-KNN keeps the label distribution, thereby retaining the overall labeling structure for each instance, whereas LDSVR takes advantage of the large margin of regression through a support vector machine.

Table 13 Experimental results on real-world datasets measured by KL-div ↓.

Dataset	LDSVR	CPNN	AA-KNN	LTSA-LE
Yeast-alpha	0.0030	0.0015	0.0018	**0.0012**
Yeast-cdc	0.0064	0.0054	0.0111	**0.0018**
Yeast-elu	0.0094	0.0085	0.0080	**0.0015**
Yeast-diau	0.0198	0.0258	0.0273	**0.0023**
Yeast-heat	0.0193	0.0187	0.0367	**0.0009**
Yeast-spo	0.0952	0.1182	0.1047	**0.0048**
Yeast-cold	0.0201	0.0219	0.0290	**0.0009**
Yeast-dtt	0.0077	0.0092	0.0096	**0.0001**
Yeast-spo5	0.0836	0.0934	0.1144	**0.0057**
Yeast-spoem	0.2484	0.1858	0.0538	**0.0003**
Human Gene	0.0355	0.0355	0.0594	**0.0285**
Natural Scene	1.4482	0.6021	0.6874	**0.5503**
Movie	0.0955	0.1190	0.0994	**0.0688**
s-JAFFE	0.0619	0.1073	0.0473	**0.0206**
s-BU_3DFE	0.0925	0.0809	0.1625	**0.0252**
Average rank	2.83	2.83	3.33	1.00

Table 14 Experimental results on real-world datasets measured by the Cosine ↑.

Dataset	LDSVR	CPNN	AA-KNN	LTSA-LE
Yeast-alpha	0.9971	0.9985	0.9981	**0.9988**
Yeast-cdc	0.9935	0.9946	0.9893	**0.9982**
Yeast-elu	0.9904	0.9913	0.9921	**0.9985**
Yeast-diau	0.9832	0.9778	0.9766	**0.9978**
Yeast-heat	0.9821	0.9826	0.9648	**0.9990**
Yeast-spo	0.9032	0.8815	0.8944	**0.9952**
Yeast-cold	0.9803	0.9785	0.9710	**0.9990**
Yeast-dtt	0.9925	0.9909	0.9903	**0.9998**
Yeast-spo5	0.9310	0.9268	0.9075	**0.9945**
Yeast-spoem	0.8293	0.8808	0.9556	**0.9997**
Human Gene	0.9647	0.9639	0.9420	**0.9717**
Natural Scene	0.4386	0.7315	0.6838	**0.7690**
Movie	**0.9371**	0.9225	0.9205	0.9364
s-JAFFE	0.9360	0.8879	0.9482	**0.9791**
s-BU_3DFE	0.9018	0.9147	0.8432	**0.9739**
Average rank	2.73	2.80	3.40	1.07

5 Conclusion

We studied a label-enhanced algorithm based on manifold. According to the optimization theory, we established the unconstrained optimization model of MSVR and provided the adaptively adjusted maximum entropy function to solve the model. The function is based on the characteristics of the original model quadratic programming problem and transforms the original objective function into an unconstrained optimization problem according to the optimization

Table 15 Experimental results on real-world datasets measured by Intersec ↑.

Dataset	LDSVR	CPNN	AA-KNN	LTSA-LE
Yeast-alpha	0.9694	0.9789	0.9771	**0.9815**
Yeast-cdc	0.9495	0.9546	0.9378	**0.9786**
Yeast-elu	0.9480	0.9488	0.9502	**0.9780**
Yeast-diau	0.9136	0.9037	0.9024	**0.9769**
Yeast-heat	0.9119	0.9154	0.8753	**0.9814**
Yeast-spo	0.8226	0.8031	0.8170	**0.9610**
Yeast-cold	0.9108	0.9001	0.8887	**0.9819**
Yeast-dtt	0.9394	0.9325	0.9444	**0.9926**
Yeast-spo5	0.8278	0.8249	0.7994	**0.9509**
Yeast-spoem	0.6631	0.7131	0.8488	**0.9875**
Human Gene	0.9071	**0.9097**	0.8567	0.9095
Natural Scene	0.3522	0.5683	0.5008	**0.5791**
Movie	0.8203	0.7973	0.8183	**0.8479**
s-JAFFE	0.8603	0.7990	0.8817	**0.9102**
s-BU_3DFE	0.8210	0.8255	0.7771	**0.9048**
Average rank	2.93	2.80	3.20	1.07

principle. To avoid the numerical overflow phenomenon in the standard maximum entropy function, we proposed an MSVR model for adaptively adjusting the Shannon entropy function.

We state the convergence of the model and solve it with the quasi-Newton method, which is particularly suitable for problems with a large number of sample points. Extensive experimental results using real-world multilabel datasets show that the algorithm can effectively improve the training speed and prediction accuracy of multilabel datasets.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 61702270, 41971343, and 61702271), China Postdoctoral Science Foundation (No. 2017M621592), and China Scholarship Council (No. CSC201906865006).

References

[1] X. Geng, N. Xu, and R. Shao, Label enhancement for label distribution learning, *Journal of Computer Research and Development*, vol. 54, no. 6, pp. 1171–1184, 2017.

[2] Z. Wang, J. Xin, H. Yang, S. Tian, G. Yu, C. Xu, and Y. Yao, Distributed and weighted extreme learning machine for imbalanced big data learning, *Tsinghua Science and Technology*, vol. 22, no. 2, pp. 160–173, 2017.

[3] K. Wang, M. Yang, W. Yang, and Y. Yin, Deep cross-view label embedding with correlation and structure preserved for multi-label classification, in *Proc. of International Conference on Tools with Artificial Intelligence*, Volos, Greece, 2018, pp. 12–19.

[4] F. Tai and H. Lin, Multilabel classification with principal label space transformation, *Neural Computation*, vol. 24, no. 9, pp. 2508–2542, 2012.

[5] L. Sun, S. Ji, and J. Ye, Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 194–200, 2011.

[6] D. Tuia, J. Verrelst, L. Alonso, F. Prez-Cruz, and G. Camps-Valls, Multioutput support vector regression for remote sensing biophysical parameter estimation, *Geoscience and Remote Sensing Letters*, vol. 8, no. 4, pp. 804–808, 2011.

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

[8] P. Hou, X. Geng, and M. Zhang, Multi-label manifold learning, in *Proc. of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, AZ, USA, 2016, pp. 1680–1686.

[9] X. Zhu, J. Lafferty, and R. Rosenfeld, Semi-supervised learning with graphs, PhD dissertation, Dept. Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 2005.

[10] N. Xu, A. Tao, and X. Geng, Label enhancement for label distribution learning, in *Proc. of the 27th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 2018, pp. 2926–2932.

[11] C. Peng, A. Tao, and X. Geng, Label embedding based on multi-scale locality preservation, in *Proc. of the 27th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 2018, pp. 2623–2629.

[12] R. Shao, N. Xu, and X. Geng, Multi-label learning with label enhancement, in *Proc. of IEEE International Conference on Data Mining*, Singapore, 2018, pp. 437–446.

[13] Z. Zhang and H. Zha, Principal manifolds and nonlinear dimensionality reduction via tangent space alignment, *SIAM Journal of Scientific Computing*, vol. 26, no. 1, pp. 313–338, 2004.

[14] Y. Jiang, Z. Deng, J. Wang, P. Qian, and S. Wang, Collaborative partition multi-view fuzzy clustering algorithm using entropy weighting, *Journal of Software,* vol. 25, no. 10, pp. 2293–2311, 2014.

[15] M. Zhang and Z. Zhou, Multi-label neural networks with applications to functional genomics and text categorization, *IEEE Transactions on Knowledge and Data Engineering,* vol. 18, no. 10, pp. 1338–1351, 2006.

[16] X. Geng and P. Hou, Pre-release prediction of crowd opinion on movies by label distribution learning, in *Proc. of the 24th International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, 2015, pp. 3511–3517.

[17] X. Geng, C. Yin, and Z. 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.

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

[19] J. Demsar, Statistical comparisons of classifiers over multiple datasets, *Journal of Machine Learning Research,* vol. 7, no. 1, pp. 1–30, 2006.

Chao Tan received the BEng and MEng degrees from Southeast University in 2005 and 2009, respectively, and received the PhD degree from Tongji University in 2015. She is now a postdoctoral researcher in Southeast University supervised by Prof. Xin Geng. She joined Nanjing Normal University as a lecturer in 2015 and is an associate professor at present. Her research interests generally focus on machine learning, multi-label manifold learning, and data mining.

Genlin Ji received the BEng and MEng degrees from Nanjing University of Aeronautics and Astronautics in 1986 and 1989, respectively, and received the PhD degree from Southeast University in 2004. He is now a professor in Nanjing Normal University. His research interests generally focus on data mining and its application.

Richen Liu received the BEng and MEng degrees from Sichuan University in 2008 and 2011, respectively, and the PhD degree from Peking University in 2017. Now he is a lecturer in Nanjing Normal University. He is the author/coauthor of more than 20 international conference/journal papers. His research interests include data visualization and machine learning.

Yanqiu Cao received the BEng degree from Yangzhou University in 2017. She is now a master student in Nanjing Normal University. Her research interests generally focus on data mining and machine learning.