



Joint label-specific features and label correlation for multi-label learning with missing label

Ziwei Cheng¹ · Ziwei Zeng¹

Published online: 8 July 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Existing multi-label learning classification algorithms ignore that class labels may be determined by some features in the original feature space. And only a partial label of each instance can be obtained for some real applications. Therefore, we propose a novel algorithm named joint Label-Specific features and Label Correlation for multi-label learning with Missing Label (LSLC-ML) and its optimized version to solve the above-mentioned problems. First, a missing label can be recovered by the learned positive and negative label correlations from the incomplete training data sets, then the label-specific features can be selected, finally the multi-label classification task can be modeled by combining the labelspecific feature selections, missing labels and positive and negative label correlations. The experimental results show that our algorithm LSLC-ML has strong competitiveness compared with some state-of-the-art algorithms in evaluation matrices when tested on benchmark multi-label data sets.

Keywords Missing labels · Label-specific features selections · Positive label correlations · Negative label correlations

1 Introduction

As an important branch of Artificial Intelligence, machine learning has been widely used in many practical applications, especially in the field of smart applied approaches. Many scholars have proposed many algorithms one after another, such as Zhou et al. [1], Bencherif et al. [2], Souza et al. [3], and Bae et al. [4]. These algorithms mainly use fuzzy neural networks to solve related problems. Traditional supervised learning methods are often only aimed at a single label, but in reality, an object often has multiple classification labels. This type of supervised learning is called multi-label classification. At present, multi-label learning has been widely used in web page classification [5], image annotation [6], biological analysis [7] and other fields.

There are two ways to solve multi-label classification algorithm: problem transformation and algorithm

adaptation. The idea of problem transformation is to transform multi-label classification into a series of single-label classifications, such as MLSVM [6] (Multi-label Support Vector Machine). However, MLSVM does not take the label correlations into account. Different from the problem transformation, the idea of algorithm adaptation is to adapt the existing single-label classification algorithm to solve the multi-label classification, such as ML-KNN [8] (Multi-Label K Nearest Neighbor), it introduces MAP (Maximum A Posteriori Probability Estimate) based on the traditional KNN algorithm, but the ML-KNN ignores the label correlations. Therefore, many algorithms related to label correlations have been proposed one after another, such as [9–14] etc. In fact, there is also a mutual exclusion phenomenon between labels, which are called negative correlations. For example, both Huang et al. [15] and Wu et al. [16] are linking the positive and negative label correlations. There are also some scholars who put forward some corresponding methods to solve the problem of class imbalance in multi-label learning, such as [17–19].

In the classification task, sometimes the curse of dimensionality often occurs because the data features are too large and feature selection is a method to solve the dimension problem. Up to now, a lot of algorithms have been proposed, which based on the assumption that selected features have something to do with all class labels, such

✉ Ziwei Zeng
zzwustl@163.com

Ziwei Cheng
chengzwzy@163.com

¹ University of Science and Technology Liaoning, Anshan, China

as [20–24] etc. However, each class label might be determined by some specific feature subsets. For example, in some text classification scenarios, if there are a series of vocabulary such as “basketball” and “ping-pong ball” in the layout and then there will be a greater probability that the layout will be judged as “sport”. These words are called label-specific features. In order to solve the problems mentioned above, many new algorithms are proposed to select label-specific features [25–27].

Unfortunately, many state-of-the-art methods have a poor performance when the labels are missing in some applications. According to previous research, it is inappropriate that many multi-label learning classification algorithms based on the assumption that all class labels are observed. But in many practical application, it is difficult to obtain complete labels.

To solve missing label problem in classification tasks, there are many algorithm have been proposed, such as [28–33], and it is feasible to deal with missing label. However, these algorithms use identical data representation composed of all the features in the discrimination of all the class labels. Therefore, how to recover missing label by the learned positive and negative label correlations from the incomplete training data sets of each different class label and select label-specific features with it remains a hot spots of multi-label learning with missing labels. The motivation of this article is summarized as follows.

- The class labels may be determined by some features in the original feature space.
- Many algorithms only deal with the missing labels by the positive label correlations and ignore the negative label correlations.

To solve the aforesaid two problems, we proposed a novel algorithm which jointly label-Specific features selections and recover missing labels for multi-label learning by using label correlation in a unified model. Compared with other algorithms, the innovation of the algorithm proposed in this paper is that when completing the missing label matrix, we not only use the positive correlation of labels, but also use the negative correlation of labels. The main contributions are summarized as follows.

- This is the first model that combines label-specific features selections, missing label and label correlations of positive and negative.
- The positive and negative label correlations can be learned directly from the incomplete training datasets,

and used to complete the matrix and multi-label classification model can be modeled by selecting label-specific features iteratively.

- Our proposed algorithm performs competitive performance against existing state-of-the-art algorithm from the experiment in this paper.

The rest of this paper is organized as follows. Section 2 reviews the related work of labelspecific features, missing labels and label correlations. In Section 3, we propose two versions of the algorithm. Section 4 gives the optimization of the proposed algorithm. The experimental setup and conclusion are shown in Section 5 and Section 6 respectively.

2 Related work

2.1 Multi-label learning with label-specific feature

In multi-label learning, it is useful to the classification results if the label-specific feature can be found. Zhang et al. [25] proposed LIFT, and it calculates the distance between the original data and the cluster center by clustering of positive and negative instances of each class label respectively, and takes the distance as part of the original data features. Similarity, LSDM [34] utilizes clustering algorithm by setting different ratio parameter for positive and negative instances class label respectively, and finds label-specific feature, which consists of distance information and spatial topology information. However, they does not consider label correlations. Comparatively speaking, LSF-CI [35] selects label-specific feature for each class label by modeling label correlations in label space and feature space simultaneously.

Huang et al. [26] and Weng et al. [36] proposed LLSF-DL and LF-LPLC respectively, the former joints label-specific features and class-dependent labels by stacking sparsely, while the latter joints label-specific features and local pairwise label correlations. Therefore, although these algorithms can deal with label-specific feature, they do not perform particularly well when the label is missing. There are also some scholars who consider this issue from other angles, such as [37, 38]. NSLSF [37] translates the logic labels to the numerical ones and utilize the numerical labels for extracting label specific features. FSSL [38] selects labelspecific features for each newly-arrived label with designing inter-class discrimination and intra-class neighbor recognition.

2.2 Multi-label learning with missing labels

It is inappropriate to assume that all labels are observed. Therefore, there are many algorithms for missing labels, such as [28–30]. Wu et al. [28] proposed a new model for multi-label with missing label and it has been proved that this model can solve the problem of label missing. Maxide [29] speedups matrix completion by exploring the side information and it is assumed that the matrix to be supplemented and the side information matrix are shared the latent information. SVMNN [30] is an algorithm based on the SVM method, it constructs two graph to explore sample smoothness and class smoothness respectively, and then the loss function is modeled which combines these two factors. The algorithm mentioned just now is aimed at missing labels for each instance is unknown, so [32, 33] are proposed based on which partial missing labels are known. SSWL [33] is a semi-supervised learning to supplement missing label by both instance similarity and label similarity are considered. iMVML [32] explores a shared subspace that handles multi-view multi-label learning, and the subspace can be learned from incomplete views with weak labels, label correlations and a predictor in this subspace simultaneously. Besides, there are some frameworks that can handle label correlations and missing label or others simultaneously, such as [31, 39]. LSML [39] can handle multi-label task with missing label, label correlations and label-specific feature selection and it is similar to our method in this paper that select label-specific feature by adding l_1 regularization. MLMLFS [31] jointly deals with feature selections with missing label and models classifier in a unified framework. It is modeled by $l_{2,p}$ regularization to select the most discriminative features and a manifold regularization to explore label correlations. That is to say, how to find the label correlations is the most critical issue.

2.3 Multi-label learning with label correlations

Whatever it is missing labels or feature selections, the label correlation plays an important role in it and it has been proved to improve the classification ability of the multi-label learning algorithm. Examples are ML-LOC [10], Glocal [13] etc. ML-LOC can obtain local label correlations by clustering and it is taken into account in the feature space, while Glocal can obtain global and local label correlation based on ML-LOC. However, these label correlations are only positive label correlations. There is also a mutual exclusion between labels, called negative correlations, such as RAKEL [12], LPLC [15] and CPNL

[16]. LPLC can explore positive and negative label correlations matrix during the training process by calculating the max posterior probability of each class label given by another one on nearest neighbors sets. Similarity, RAKEL is an improved version of [14], it adds positive and negative correlations into k-labelsets by calculating the cosine similarity of the two matrices. Nevertheless, CPNL does not obtain positive and negative label correlation through calculation, but is modeled in a unified framework. In conclusion, it is proposed to joint label-specific features selections and label Correlation for Multi-label Learning with missing label in this paper.

3 The proposed model

3.1 Notations

For an arbitrary matrix \mathbf{W} , \mathbf{W}^i and \mathbf{W}^j denote the i -th row and j -th column of \mathbf{W} , \mathbf{W}^T denote the deposition of \mathbf{W} . In multi-label learning, given dataset $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, where we denote the feature vector of the i -th instance by $\mathbf{x}_i \in R^m$, and denote the i -th ground-truth label vector by $\mathbf{y}_i \in \{0, 1\}^l$, and m represents the number of features of the instance and l represents the number of class labels associated with an instance. For simplicity, the training data is expressed as matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T \in R^{n \times m}$, $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n]^T \in \{0, 1\}^{n \times l}$, where $y_{il} = 1$ represents the i -th instance is associated with the l -th label, and $y_{il} = 0$ represents the i -th instance is nothing to do with the l -th label or the l -th label is missing.

3.2 Label-specific feature selection

Given datasets, we propose the assumption that each class label is only determined by a certain subset of specific features from the original feature space. In order to select this feature, that is, label-specific feature, l_1 regularization is used in the modeling of a linear model. Naturally, label-specific feature is selected by no-zero values of each $\mathbf{w}^j \in R^m$, the optimization problem is defined as

$$\min_{\mathbf{W}} \frac{1}{2} \|\mathbf{XW} - \mathbf{Y}\|_F^2 + \lambda_1 \|\mathbf{W}\|_1 \tag{1}$$

where $\mathbf{W} \in R^{m \times l}$ is the regression coefficient and $\mathbf{W} = [\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^l]$, and λ_1 is the tradeoff parameter. $\|\cdot\|_F^2$ is the summing the squares of each element in matrix \cdot .

3.3 Learning multi-label with missing label

In previous studies, we assume that the labels of the given training data sets are complete and all are observable. However, this assumption is often not true. Considering that labeling is a huge project, missing labels have become a hot spot in multi-label learning. In order to solve such problems, it is proposed to a method that recovers incomplete label matrix to complete label matrix by making full use of the label correlations. First of all, we assume that any unobserved, that is, missing labels can be explored by the relationship between observed labels. Let $\mathbf{P} \in R^{l \times l}$ be label correlations, where p_{ij} represents the degree of correlation between the i -th label and the j -th label. Following the idea of Huang et al. [39], we assume that one class label may be correlated with only a subset of class labels, thus we add the l_1 -norm regularizer over \mathbf{P} . The optimization problem can be rewritten as

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{P}} \quad & \frac{1}{2} \|\mathbf{XW} - \mathbf{YP}\|_F^2 + \lambda_1 \|\mathbf{W}\|_1 + \lambda_2 \|\mathbf{P}\|_1 + \\ & \lambda_4 \|\mathbf{YP} - \mathbf{Y}\|_F^2 \end{aligned} \quad (2)$$

s.t. $\mathbf{P} \geq 0$

3.4 Learning positive label correlations

In the multi-label classification task, the accuracy of classification can be greatly improved if the relationship between labels can be considered. Therefore, we incorporate the label pairwise correlation by calculating the Euclidean distance between any pair of coefficient vectors which are constrained by the learned label correlation matrix \mathbf{P} . Here, for two label y_i and y_j have positive correlations, that is, p_{ij} or p_{ji} is large, and their label-specific features have more probability to be same. That is to say, the corresponding regression coefficient \mathbf{w}^i and \mathbf{w}^j will be more similar, and vice versa. According to what has been mentioned above, we can add the regularizer by the Euclidean distance of \mathbf{w}^i and \mathbf{w}^j as follows,

$$\begin{aligned} \text{POS} &= \frac{1}{2} \sum_{i,j=1}^l p_{ij} \|\mathbf{w}^i - \mathbf{w}^j\|_2^2 \\ &= \frac{1}{2} \left(\sum_{i,j=1}^l p_{ij} (\|\mathbf{w}^i\|_2^2 + \|\mathbf{w}^j\|_2^2) - 2 \sum_{i,j=1}^l p_{ij} \langle \mathbf{w}^i, \mathbf{w}^j \rangle \right) \\ &= \text{tr}(\mathbf{W} \mathbf{D}_p \mathbf{W}^T) - \text{tr}(\mathbf{W} \mathbf{P} \mathbf{W}^T) \\ &= \text{tr}(\mathbf{W} \mathbf{L}_p \mathbf{W}^T) \end{aligned} \quad (3)$$

Where $\mathbf{L}_p = \mathbf{D}_p - \mathbf{P} \in R^{l \times l}$ is the laplacian matrix and \mathbf{D}_p is the diagonal matrix with $\mathbf{P}\mathbf{1}$, where $\mathbf{1}$ is a matrix with all values one. Thus, the optimization problem can be rewritten as,

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{P}} \quad & \frac{1}{2} \|\mathbf{XW} - \mathbf{YP}\|_F^2 + \lambda_1 \|\mathbf{W}\|_1 + \lambda_2 \|\mathbf{P}\|_1 \\ & + \lambda_4 \|\mathbf{YP} - \mathbf{Y}\|_F^2 + \lambda_5 \text{tr}(\mathbf{W} \mathbf{L}_p \mathbf{W}^T) \end{aligned} \quad (4)$$

s.t. $\mathbf{P} \geq 0$

3.5 Learning negative label correlations

In fact, the label relation mentioned above be actually a positive label correlation. There is also a mutual exclusion between labels, called negative label correlations [16]. For example, for "aquarium", the relationship between "seal" and "aquarium" will be relatively close, but the relationship with "tiger" will be relatively weak. Thus, similar to the idea of positive correlation of labels, we can learned negative label correlation matrix \mathbf{N} , for two label y_i and y_j have negative correlations, that is, n_{ij} or n_{ji} is large, and their label-specific features have more probability to be same. That is to say, the corresponding regression coefficient \mathbf{w}^i and \mathbf{w}^j will be more similar, and vice versa. Note, the Euclidean distance of \mathbf{w}^i and \mathbf{w}^j should be large. Therefore, we can add the regularizer as follows,

$$\begin{aligned} \text{NEG} &= \frac{1}{2} \sum_{i,j=1}^l n_{ij} \|\mathbf{w}^i - (-\mathbf{w}^j)\|_2^2 \\ &= \frac{1}{2} \left(\sum_{i,j=1}^l n_{ij} (\|\mathbf{w}^i\|_2^2 + \|\mathbf{w}^j\|_2^2) + 2 \sum_{i,j=1}^l n_{ij} \langle \mathbf{w}^i, \mathbf{w}^j \rangle \right) \\ &= \text{tr}(\mathbf{W} \mathbf{D}_n \mathbf{W}^T) + \text{tr}(\mathbf{W} \mathbf{N} \mathbf{W}^T) \\ &= \text{tr}(\mathbf{W} \mathbf{L}_n \mathbf{W}^T) \end{aligned} \quad (5)$$

where $\mathbf{L}_n = \mathbf{D}_n + \mathbf{N} \in R^{l \times l}$ and \mathbf{D}_n is the diagonal matrix with $\mathbf{N}\mathbf{1}$, where $\mathbf{1}$ is a matrix with all values one. Note that we propose a hypothesis that the relationship between labels is a combination of positive label correlation and negative label correlation. For simplicity, we adopt linear combination here, that is, $\mathbf{P} + \mathbf{N}$. Thus, the final optimization problem can be rewritten as,

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{P}, \mathbf{N}} \quad & \frac{1}{2} \|\mathbf{XW} - \mathbf{Y}(\mathbf{P} + \mathbf{N})\|_F^2 + \lambda_1 \|\mathbf{W}\|_1 + \lambda_2 \|\mathbf{P}\|_1 + \lambda_3 \|\mathbf{N}\|_1 + \\ & \lambda_4 \|\mathbf{Y}(\mathbf{P} + \mathbf{N}) - \mathbf{Y}\|_F^2 + \lambda_5 \text{tr}(\mathbf{W} \mathbf{L}_p \mathbf{W}^T) + \lambda_6 \text{tr}(\mathbf{W} \mathbf{L}_n \mathbf{W}^T) \end{aligned} \quad (6)$$

s.t. $\mathbf{P} \geq 0 \quad \mathbf{N} \geq 0$

4 Model optimization and prediction

In this section, considering that object function is non-smooth, the accelerated proximal gradient method is applied in this

issue according to paper [39]. For simplicity, let Φ be model parameters, that is, $\Phi = (\mathbf{W}, \mathbf{P}, \mathbf{N})$. Thus, object function(6) can be written as,

$$\min_{\Phi} \{F(\Phi) \equiv f(\Phi) + g(\Phi)\} \tag{7}$$

Where

$$f(\Phi) = \frac{1}{2} \|\mathbf{XW} - \mathbf{Y}(\mathbf{P} + \mathbf{N})\|_F^2 + \lambda_4 \|\mathbf{Y}(\mathbf{P} + \mathbf{N}) - \mathbf{Y}\|_F^2 + \lambda_5 \text{tr}(\mathbf{WL}_P \mathbf{W}^T) + \lambda_6 \text{tr}(\mathbf{WL}_N \mathbf{W}^T) \tag{8}$$

$$g(\Phi) = \lambda_1 \|\mathbf{W}\|_1 + \lambda_2 \|\mathbf{P}\|_1 + \lambda_3 \|\mathbf{N}\|_1 \tag{9}$$

Note, although function $f(\Phi)$ and $g(\Phi)$ are convex, the latter is not smooth. First of all, we use the following function defined in paper [40].

$$Q_L(\Phi, \Phi^i) = f(\Phi^i) + \langle \nabla f(\Phi), \Phi - \Phi^i \rangle + \frac{L}{2} \|\Phi - \Phi^i\|_F^2 + g(\Phi) \tag{10}$$

Here, for any $L \geq L_f$, $Q_L(\Phi, \Phi^i) \geq F(\Phi)$ will always be found, where L_f is the Lipschitz constant. Next, minimizing a sequence of separable quadratic approximations to $F(\Phi)$ by the proximal algorithm. Similar to paper [40], we can solve Φ by minimizing $Q_L(\Phi)$, the calculation formula is as follows,

$$\begin{aligned} \Phi^* &= \arg \min_{\Phi} Q_L(\Phi, \Phi^i) \\ &= \arg \min_{\Phi} g(\Phi) + \frac{L}{2} \|\Phi - G^i\|_F^2 \end{aligned} \tag{11}$$

where

$$\mathbf{G}^i = \Phi^i - \frac{1}{L} \nabla f(\Phi^i) \tag{12}$$

It is effective that speed up the convergence by set $\Phi^i = \Phi + \frac{\alpha_i - 1}{\alpha_i} (\Phi_i - \Phi_{i-1})$ for sequence α_i , where α_i satisfies $\alpha_{i+1}^2 - \alpha_{i+1} \leq \alpha_i^2$, and Φ_i is the i -th iteration of Φ .

4.1 Update W

Firstly, Updating \mathbf{W} with fixed \mathbf{P} and \mathbf{N} , the derivation of function (8) can be calculated as follows,

$$\begin{aligned} \nabla_{\mathbf{W}} f(\Phi) &= \mathbf{X}^T \mathbf{XW} - \mathbf{X}^T \mathbf{Y} \mathbf{P} - \mathbf{X}^T \mathbf{Y} \mathbf{N} + \lambda_5 \mathbf{W} (\mathbf{L}_p + \mathbf{L}_p^T) \\ &\quad + \lambda_6 \mathbf{W} (\mathbf{L}_n + \mathbf{L}_n^T) \end{aligned} \tag{13}$$

Then, according to Eq. (11), \mathbf{W} can be updated by

$$\mathbf{W}^i = \mathbf{W}_i + \frac{\alpha_{i-1} - 1}{\alpha_i} (\mathbf{W}_i - \mathbf{W}_{i-1}) \tag{14}$$

$$\mathbf{W}^{i+1} = \text{prox}_{\epsilon} \left(\mathbf{W}^i - \frac{1}{L} \nabla_{\mathbf{W}} f(\mathbf{W}^i, \mathbf{P}, \mathbf{N}) \right) \tag{15}$$

Where, ϵ is step size. Here, because \mathbf{W} in $g(\Phi)$ is regularization term with l_1 -norm, the element-wise soft-threshold operator is adopted in solving this value. The calculation formula is as follows,

$$\text{prox}_{\epsilon}(w_{ij}) = (|w_{ij}| - \epsilon)_+ \text{sign}(w_{ij}) \tag{16}$$

where $(\cdot)_+ = \max(\cdot, 0)$.

4.2 Update P

Similarly, Updating \mathbf{P} with fixed \mathbf{W} and \mathbf{N} , the derivation of function (8) can be calculated as follows,

$$\begin{aligned} \nabla_{\mathbf{P}} f(\Phi) &= (1 + \lambda_4) \mathbf{Y}^T \mathbf{Y} \mathbf{P} \\ &\quad + (1 + \lambda_4) \mathbf{Y}^T \mathbf{Y} \mathbf{N} - \mathbf{Y}^T \mathbf{XW} - \lambda_4 \mathbf{Y}^T \mathbf{Y} \end{aligned} \tag{17}$$

Thus, \mathbf{P} can be updated by

$$\mathbf{P}^i = \mathbf{P}_i + \frac{\alpha_{i-1} - 1}{\alpha_i} (\mathbf{P}_i - \mathbf{P}_{i-1}) \tag{18}$$

$$\mathbf{P}^{i+1} = \text{prox}_{\epsilon} \left(\mathbf{P}^i - \frac{1}{L} \nabla_{\mathbf{P}} f(\mathbf{W}, \mathbf{P}^i, \mathbf{N}) \right) \tag{19}$$

Where, ϵ is step size. Here, because \mathbf{P} in $g(\Phi)$ is nonnegative regularization term with l_1 -norm, the element-wise soft-threshold operator is adopted in solving this value. The calculation formula is as follows,

$$\text{prox}_{\epsilon}(p_{ij} - \epsilon) = (|p_{ij}| - \epsilon)_+ \tag{20}$$

where $(\cdot)_+ = \max(\cdot, 0)$.

4.3 Update N

Similarly, Updating \mathbf{N} with fixed \mathbf{W} and \mathbf{P} , the derivation of function (8) can be calculated as follows,

$$\begin{aligned} \nabla_{\mathbf{N}} f(\Phi) &= (1 + \lambda_4) \mathbf{Y}^T \mathbf{Y} \mathbf{P} \\ &\quad + (1 + \lambda_4) \mathbf{Y}^T \mathbf{Y} \mathbf{N} - \mathbf{Y}^T \mathbf{XW} - \lambda_4 \mathbf{Y}^T \mathbf{Y} \end{aligned} \tag{21}$$

Thus, \mathbf{N} can be updated by

$$\mathbf{N}^i = \mathbf{N}_i + \frac{\alpha_{i-1}-1}{\alpha_i} (\mathbf{N}_i - \mathbf{N}_{i-1}) \tag{22}$$

$$\mathbf{N}^{i+1} = \mathbf{prox}_\delta \left(\mathbf{N}^i - \frac{1}{L} \nabla_{\mathbf{N}} f(\mathbf{W}, \mathbf{P}, \mathbf{N}^i) \right) \tag{23}$$

where, ε is step size. Here, because \mathbf{N} in $g(\Phi)$ is nonnegative regularization term with l_1 -norm, the element-wise soft-threshold operator is adopted in solving this value. The calculation formula is as follows,

$$\mathbf{prox}_\varepsilon(n_{ij} - \varepsilon) = (|n_{ij}| - \varepsilon)_+ \tag{24}$$

where $(\cdot)_+ = \max(\cdot, 0)$. Note, the derivative of the function(8) with respect to \mathbf{P} is the same as the derivative of \mathbf{N} , because we assume that the label relationship is a simplest linear combination of \mathbf{P} and \mathbf{N} , that is, $\mathbf{P} + \mathbf{N}$. All optimization steps are summarized in algorithm1, called LSLC for simplicity.

Algorithm 1 Optimization of LSLC

Input: Training Data \mathbf{X} , Training Label \mathbf{Y} ;

Output: $\mathbf{W}; \mathbf{P}; \mathbf{N}$;

1: **initialize**

model parameters $\mathbf{W}; \mathbf{P}; \mathbf{N}$;

tradeoff parameters $\lambda_1; \lambda_2; \lambda_3; \lambda_4; \lambda_5; \lambda_6$;

iteration parameters $\alpha_1; \alpha_2; i$;

relative parameters $\mathbf{L}_p; \mathbf{L}_n$;

2: **repeat**

3: calculate \mathbf{L}_p and \mathbf{L}_n ;

4: calculate L_f according to (26) ;

5: calculate \mathbf{W}^i by (14) ;

6: $\mathbf{G}_w^i \leftarrow \mathbf{W}^i - \frac{1}{L} \nabla_{\mathbf{W}} f(\mathbf{W}^i, \mathbf{P}_i, \mathbf{N}_i)$

7: $\mathbf{W}_{i+1} \leftarrow \mathbf{prox}_\delta(\mathbf{G}_w^i)$ by (16), where $\delta = \frac{\lambda_1}{L_f}$

8: calculate \mathbf{P}^i by (18) ;

9: $\mathbf{G}_p^i \leftarrow \mathbf{W}^i - \frac{1}{L} \nabla_{\mathbf{P}} f(\mathbf{W}_i, \mathbf{P}^i, \mathbf{N}_i)$

10: $\mathbf{P}_{i+1} \leftarrow \mathbf{prox}_\delta(\mathbf{G}_p^i)$ by (20), where $\delta = \frac{\lambda_2}{L_f}$

11: calculate \mathbf{N}^i by (22) ;

12: $\mathbf{G}_n^i \leftarrow \mathbf{W}^i - \frac{1}{L} \nabla_{\mathbf{N}} f(\mathbf{W}_i, \mathbf{P}_i, \mathbf{N}^i)$

13: $\mathbf{N}_{i+1} \leftarrow \mathbf{prox}_\delta(\mathbf{G}_n^i)$ by (24), where $\delta = \frac{\lambda_3}{L_f}$

14: $\alpha_{i+1} \leftarrow \frac{1 + \sqrt{4\alpha_i^2 + 1}}{2}$

15: $i \leftarrow i + 1$

16: **util converge**

17: $\mathbf{W}^* = \mathbf{W}_i, \mathbf{P}^* = \mathbf{P}_i, \mathbf{N}^* = \mathbf{N}_i$

18: **return** $\mathbf{W}^*, \mathbf{P}^*, \mathbf{N}^*$

Algorithm 2 Optimization of LSLCnew

Input: Training Data \mathbf{X} , Training Label \mathbf{Y} ;

Output: $\mathbf{W}; \mathbf{P}; \mathbf{N}$;

1: **initialize**

model parameters $\mathbf{W}; \mathbf{P}; \mathbf{N}$;

tradeoff parameters $\lambda_1; \lambda_2; \lambda_3; \lambda_4; \lambda_5; \lambda_6$;

iteration parameters $\alpha_1; \alpha_2; i$;

relative parameters $\mathbf{L}_p; \mathbf{L}_n$;

2: **repeat**

3: **while** $Q_L(\Phi, \Phi^i) > F(\Phi)$ **do**

4: $L = \zeta L$, where ζ is the search step size ;

5: calculate \mathbf{W}^i by (14) ;

6: $\mathbf{G}_w^i \leftarrow \mathbf{W}^i - \frac{1}{L} \nabla_{\mathbf{W}} f(\mathbf{W}^i, \mathbf{P}_i, \mathbf{N}_i)$

7: $\mathbf{W}_{i+1} \leftarrow \mathbf{prox}_\delta(\mathbf{G}_w^i)$ by (16), where $\delta = \frac{\lambda_4}{L_f}$

8: calculate \mathbf{P}^i by (18) ;

9: $\mathbf{G}_p^i \leftarrow \mathbf{W}^i - \frac{1}{L} \nabla_{\mathbf{P}} f(\mathbf{W}_i, \mathbf{P}^i, \mathbf{N}_i)$

10: $\mathbf{P}_{i+1} \leftarrow \mathbf{prox}_\delta(\mathbf{G}_p^i)$ by (20), where $\delta = \frac{\lambda_5}{L_f}$

11: calculate \mathbf{N}^i by (22) ;

12: $\mathbf{G}_n^i \leftarrow \mathbf{W}^i - \frac{1}{L} \nabla_{\mathbf{N}} f(\mathbf{W}_i, \mathbf{P}_i, \mathbf{N}^i)$

13: $\mathbf{N}_{i+1} \leftarrow \mathbf{prox}_\delta(\mathbf{G}_n^i)$ by (24), where $\delta = \frac{\lambda_6}{L_f}$

14: $\alpha_{i+1} \leftarrow \frac{1 + \sqrt{4\alpha_i^2 + 1}}{2}$

15: $i \leftarrow i + 1$

16: **util converge**

17: $\mathbf{W}^* = \mathbf{W}_i, \mathbf{P}^* = \mathbf{P}_i, \mathbf{N}^* = \mathbf{N}_i$

18: **return** $\mathbf{W}^*, \mathbf{P}^*, \mathbf{N}^*$

Algorithm 3 Model Prediction

Input: Model Parameter \mathbf{W} , Test Data \mathbf{D}_{ts} ;

Output: Predictive label \mathbf{Y}_{ts} ;

1. calculate $Sc = \mathbf{W} \mathbf{D}_{ts}$;

2. $\mathbf{Y}_{ts} = \text{Sign}(Sc - \tau)$;

3. **return** \mathbf{Y}_{ts}

4.4 Lipschitz constant

Given model parameters $\Phi_1 = (\mathbf{W}_1, \mathbf{P}_1, \mathbf{N}_1)$ and $\Phi_2 = (\mathbf{W}_2, \mathbf{P}_2, \mathbf{N}_2)$, then, we have

$$\begin{aligned}
 \|\nabla f(\Phi_1) - \nabla f(\Phi_2)\|_F^2 &= \left\| \begin{array}{c} \mathbf{X}^T \mathbf{X} \Delta \mathbf{W} - \mathbf{X}^T \mathbf{Y} \Delta \mathbf{P} - \mathbf{X}^T \mathbf{Y} \Delta \mathbf{N} + \lambda_5 \Delta \mathbf{W} (\mathbf{L}_p + \mathbf{L}_p^T) + \lambda_6 \Delta \mathbf{W} (\mathbf{L}_n + \mathbf{L}_n^T) \\ (1 + \lambda_4) \mathbf{Y}^T \mathbf{Y} \Delta \mathbf{P} + (1 + \lambda_4) \mathbf{Y}^T \mathbf{Y} \Delta \mathbf{N} - \mathbf{Y}^T \mathbf{X} \Delta \mathbf{W} \\ (1 + \lambda_4) \mathbf{Y}^T \mathbf{Y} \Delta \mathbf{P} + (1 + \lambda_4) \mathbf{Y}^T \mathbf{Y} \Delta \mathbf{N} - \mathbf{Y}^T \mathbf{X} \Delta \mathbf{W} \end{array} \right\|_F^2 \\
 &= \left\| \mathbf{X}^T \mathbf{X} \Delta \mathbf{W} - \mathbf{X}^T \mathbf{Y} \Delta \mathbf{P} - \mathbf{X}^T \mathbf{Y} \Delta \mathbf{N} + \lambda_5 \Delta \mathbf{W} (\mathbf{L}_p + \mathbf{L}_p^T) + \lambda_6 \Delta \mathbf{W} (\mathbf{L}_n + \mathbf{L}_n^T) \right\|_F^2 + \\
 &\quad 2 \left\| (1 + \lambda_4) \mathbf{Y}^T \mathbf{Y} \Delta \mathbf{P} + (1 + \lambda_4) \mathbf{Y}^T \mathbf{Y} \Delta \mathbf{N} - \mathbf{Y}^T \mathbf{X} \Delta \mathbf{W} \right\|_F^2 \\
 &\leq 5 \left\| \mathbf{X}^T \mathbf{X} \right\|_2^2 \left\| \Delta \mathbf{W} \right\|_F^2 + 5 \left\| \mathbf{X}^T \mathbf{Y} \right\|_2^2 \left\| \Delta \mathbf{P} \right\|_F^2 + 5 \left\| \mathbf{X}^T \mathbf{Y} \right\|_2^2 \left\| \Delta \mathbf{N} \right\|_F^2 + \\
 &\quad 5 \left\| \lambda_5 (\mathbf{L}_p + \mathbf{L}_p^T) \right\|_2^2 \left\| \Delta \mathbf{W} \right\|_F^2 + 5 \left\| \lambda_6 (\mathbf{L}_n + \mathbf{L}_n^T) \right\|_2^2 \left\| \Delta \mathbf{W} \right\|_F^2 + \\
 &\quad 3 \left\| (1 + \lambda_4) \mathbf{Y}^T \mathbf{Y} \right\|_2^2 \left\| \Delta \mathbf{P} \right\|_F^2 + 3 \left\| (1 + \lambda_4) \mathbf{Y}^T \mathbf{Y} \right\|_2^2 \left\| \Delta \mathbf{N} \right\|_F^2 + 3 \left\| \mathbf{Y}^T \mathbf{X} \right\|_2^2 \left\| \Delta \mathbf{W} \right\|_F^2 \\
 &\leq \left(5 \left\| \mathbf{X}^T \mathbf{X} \right\|_2^2 + 5 \left\| \mathbf{X}^T \mathbf{Y} \right\|_2^2 + 5 \left\| \lambda_5 (\mathbf{L}_p + \mathbf{L}_p^T) \right\|_2^2 + 5 \left\| \lambda_6 (\mathbf{L}_n + \mathbf{L}_n^T) \right\|_2^2 + 3 \left\| (1 + \lambda_4) \mathbf{Y}^T \mathbf{Y} \right\|_2^2 \right) \left\| \begin{array}{c} \Delta \mathbf{W} \\ \Delta \mathbf{P} \\ \Delta \mathbf{N} \end{array} \right\|_F^2
 \end{aligned} \tag{25}$$

Thus, the Lipschitz constant of optimization problem (6) can be calculated by

$$\mathbf{L}_f = \sqrt{5 \left\| \mathbf{X}^T \mathbf{X} \right\|_2^2 + 5 \left\| \mathbf{X}^T \mathbf{Y} \right\|_2^2 + 5 \left\| \lambda_5 (\mathbf{L}_{p'}) \right\|_2^2 + 5 \left\| \lambda_6 (\mathbf{L}_{n'}) \right\|_2^2 + 3 \left\| (1 + \lambda_4) \mathbf{Y}^T \mathbf{Y} \right\|_2^2} \tag{26}$$

where $\mathbf{L}_{p'} = \mathbf{L}_p + \mathbf{L}_p^T$ and $\mathbf{L}_{n'} = \mathbf{L}_n + \mathbf{L}_n^T$.

4.5 Time complex

In this part, let us briefly analyze the time complexity of the optimization algorithm. For matrix $\mathbf{X} \in R^{n \times m}$, $\mathbf{Y} \in \{0, 1\}^{n \times l}$, $\mathbf{W} \in R^{m \times l}$, $\mathbf{P} \in R^{l \times l}$, $\mathbf{N} \in R^{l \times l}$, $\mathbf{L} \in L^{l \times l}$, where n denotes the number of instances, l denotes the number of labels for corresponding instances, and m denotes the dimensions of instances. In LSLC-1, the positive and negative label correlations \mathbf{L}_p and \mathbf{L}_n need $O(l^2)$ in step 3, and the other key steps of the algorithm lie in the calculation of the Lipschitz constant and the derivation of the matrix. In step 4, the calculation of the Lipschitz constant needs $O(m^3 + l^3 + mnl)$. The derivation of the matrix focuses on 6, 9 and 12 steps, the derivation of the matrix \mathbf{W} needs $O(nm^2 + ml^2 + lm^2 + mnl)$, note that the derivation of the matrix \mathbf{P} and \mathbf{N} are the similar, so the time complexity of them needs $O(nl^2 + l^3 + ml^2 + mnl)$. As mentioned above, the time complexity of the calculating the Lipschitz constant can be reduced by finding a appropriate $L \geq L_f$ [40] and it needs $O(ml^2 + m^2 n + m^2 l + nl^2 + mnl + nl + l^2 + l^3)$. These are summarized in algorithm 2, called LSLCnew for simplicity.

4.6 Model prediction

In this section, we mainly use two optimization algorithms mentioned above to obtain matrix \mathbf{W} . For unlabeled test data sets \mathbf{D}_{ts} , we can predict the label of \mathbf{D}_{ts} by calculating

the score matrix Sc and then using $Sign(Sc - \tau)$ with a given threshold τ , where $Sc = \mathbf{W} \mathbf{D}_{ts}$ and τ is set to be 0.5 in this experiment. The process of prediction is summarized in algorithm 3.

5 Experiment

5.1 Data sets

The algorithms proposed in this paper are all simulated on benchmark multi-label data sets. The specific statistical information of multi-label data sets are shown in Table 1. During

Table 1 The statistical information of data sets

Data set	Instance	Feature	Label	Card
cal500	502	68	174	26.044
medical	978	1449	45	1.25
genbase	662	1185	27	1.252
stackex-chemistry	6961	540	175	2.11
stackex-chess	1675	585	227	2.41
stackex-cooking	10,491	577	400	2.23
stackex-cs	9270	635	274	2.56
stackex-philosophy	3971	842	233	2.27
rcv1subest1	6000	944	101	2.880
rcv1subest2	6000	944	101	2.634
rcv1subest3	6000	944	101	2.614

the simulation, for each data set, we randomly selected 80% of the data as training data and the rest as test data. The experiment is repeated ten times and a five-fold cross test is used.

5.2 Evaluation measures

In order to evaluate the performance of the experimental results, we use seven evaluation metrics. For simplicity, let t denote the number of test data, Y_i and \tilde{Y}_i denote the sets of ground-truth labels and predicted labels of the i -th instance, respectively; Z_j^P and Z_j^N denote the sets of positive and negative instance of j -th class label. $C_{x_i}(\mathbf{Y})$ is confidence score of x_i associated with label y . Thus, evaluation metrics can be computed as follows,

Hamming Loss(HL) evaluates error matching between the ground-truth labels Y_i and the predicted labels \tilde{Y}_i .

$$HL = \frac{1}{t} \sum_{i=1}^t \left(\frac{1}{t} |Y_i \Delta \tilde{Y}_i| \right) \quad (27)$$

where Δ represents the difference between the two data sets.

Subset Accuracy(SA) evaluates the accuracy between the ground-truth labels Y_i and the predicted labels \tilde{Y}_i in a more rigorous way.

$$SA = \frac{1}{t} \sum_{i=1}^t \|\mathbf{Y} = \tilde{\mathbf{Y}}\| \quad (28)$$

where $\|\cdot\|$ is an indication function.

Micro F1(MF1) is an extended version of the F1 Measure, and it takes each entry of the label vector as a single instance regardless of the distance between labels.

$$MF1 = \frac{2 \sum_{j=1}^l \sum_{i=1}^t Y_{ij} \tilde{Y}_{ij}}{\sum_{j=1}^l \sum_{i=1}^t Y_{ij} + \sum_{j=1}^l \sum_{i=1}^t \tilde{Y}_{ij}} \quad (29)$$

Average Precision(AP) evaluates the average score of related labels that are higher than the specific labels

$$AP = \frac{1}{t} \sum_{i=1}^t \frac{1}{|Y_i|} \sum_{y \in Y_i} \frac{\left\{ y' \mid \text{rank}_{C_{y'}(y')} \leq \text{rank}_{C_y(y)}, y' \in Y_i \right\}}{\text{rank}_{C_y(y)}} \quad (30)$$

Ranking Loss(RL) evaluates the fraction that the ranking of unrelated labels of the instance is lower than that of related labels.

$$RL = \frac{1}{t} \sum_{i=1}^t \left(\frac{1}{|Y_i| |\tilde{Y}_i|} \left| \left\{ (y', y'') \mid C_{x_i}(y') \leq C_{x_i}(y'') \mid (y', y'') \in Y_i \times \tilde{Y}_i \right\} \right| \right) \quad (31)$$

One Error(OE) evaluates the fraction that the highest ranking label of the instance is not correctly labeled.

$$OE = \frac{1}{t} \sum_{i=1}^t \left\| \arg \max_{y_i \in Y} C_{x_i}(y_i) \right\| \quad (32)$$

where $\|\cdot\|$ is an indication function.

Average AUC(AUC) evaluates the average fraction that the ranking of positive instance of the all class labels is higher than that of negative instance.

$$AUC = \frac{1}{l} \sum_{i=1}^l \frac{\left| \left\{ (x', x'') \mid C_{x'}(y_j) \geq C_{x''}(y_j), (x', x'') \in Z_j^P \times Z_j^N \right\} \right|}{\left| Z_j^P \right| \left| Z_j^N \right|} \quad (33)$$

For these evaluation metrics, for HL, RL and OE, the smaller the better; whereas for SA, MF1, AP and AUC, the bigger the better.

5.3 Compared algorithms

The two versions of algorithm are proposed in this paper, which are compared with the following state-of-the-art for multi-label learning algorithm. For simplicity and efficiency, all configuration for parameters in the compared algorithms are suggested in the original paper.

MLKNN [8] It introduces maximum posteriori probability (MAP) to deal with multi-label classification based on traditional KNN algorithm. The parameter k is found in $\{3, 5, \dots, 21\}$.

LPLC [15] It explores local positive and negative pairwise label correlations based on traditional KNN algorithm. The parameter k is found in $\{3, 5, \dots, 21\}$ and the tradeoff parameter α is found in $\{0.1, 0.2, \dots, 1\}$.

LIFT [25] It learns label-specific features through the idea of clustering algorithm. Here, base learner is linear kernel SVM, $r = 0.1$.

LLSF [26] It learns label-specific features. The regulation parameters λ is found in $\{2^{-10}, \dots, 2^{10}\}$.

Glocal [13] It explores both global and local label correlations when recover missing labels. The parameter $\lambda = 1$, from λ_1 to λ_5 are found in $\{2^{-5}, \dots, 2^1\}$, k is found in $\{0.1, \dots, 0.6\}$, and g is found in $\{2^{-10}, \dots, 2^{10}\}$.

LSML [39] It directly learns label-specific features with missing labels. All parameters are found in $\{2^{-5}, \dots, 2^1\}$.

LSLC and **LSLCnew** It is two versions of algorithm are proposed in this paper. It directly learns the label-specific features that missing labels while considering the positive and negative correlation of labels. All parameters are found in $\{2^{-10}, \dots, 2^1\}$.

Table 2 Experimental results of each comparing algorithm on eleven data sets in terms of HL. For HL, the smaller the better. Best results are highlighted in bold

HL	Missing rate	MLKNN	LPLC	LIFT	LLSF	Glocal	LSML	LSLC	LSLCnew
cal500	0.1	0.1393	0.1572	0.2339	0.1391	0.1510	0.1897	0.1892	0.1898
	0.2	0.1415	0.1533	0.2323	0.1425	0.1508	0.1882	0.1879	0.1884
	0.3	0.1449	0.1504	0.2323	0.1455	0.1496	0.1816	0.1856	0.1815
	0.4	0.1464	0.1472	0.2132	0.1470	0.1499	0.1753	0.1765	0.1762
	0.5	0.1488	0.1466	0.1818	0.1500	0.1483	0.1699	0.1726	0.1690
	0.6	0.1494	0.1471	0.1530	0.1499	0.1492	0.1653	0.1643	0.1643
medical	0.1	0.0167	0.0176	0.0196	0.0140	0.0275	0.0106	0.0104	0.0103
	0.2	0.0184	0.0188	0.0209	0.0150	0.0278	0.0110	0.0113	0.0108
	0.3	0.0200	0.0200	0.0220	0.0175	0.0277	0.0121	0.0125	0.0123
	0.4	0.0221	0.0217	0.0228	0.0193	0.0276	0.0130	0.0129	0.0129
	0.5	0.0244	0.0229	0.0237	0.0221	0.0276	0.0135	0.0137	0.0137
	0.6	0.0270	0.0255	0.0269	0.0254	0.0275	0.0158	0.0147	0.0152
genbase	0.1	0.0076	0.0042	0.0166	0.0064	0.0465	0.0020	0.0022	0.0022
	0.2	0.0128	0.0102	0.0157	0.0126	0.0466	0.0053	0.0052	0.0058
	0.3	0.0152	0.0162	0.0174	0.0157	0.0461	0.0090	0.0098	0.0087
	0.4	0.0281	0.0295	0.0192	0.0189	0.0462	0.0095	0.0090	0.0091
	0.5	0.0355	0.0362	0.0237	0.0227	0.0467	0.0105	0.0104	0.0109
	0.6	0.0395	0.0393	0.0388	0.0311	0.0463	0.0108	0.0111	0.0115
stackex-chemistry	0.1	0.0117	0.0118	0.0121	0.0117	0.0121	0.0158	0.0161	0.0158
	0.2	0.0118	0.0118	0.0120	0.0118	0.0121	0.0158	0.0158	0.0159
	0.3	0.0119	0.0118	0.0120	0.0118	0.0120	0.0154	0.0159	0.0154
	0.4	0.0120	0.0119	0.0120	0.0120	0.0121	0.0152	0.0154	0.0153
	0.5	0.0120	0.0119	0.0120	0.0121	0.0121	0.0148	0.0151	0.0150
	0.6	0.0121	0.0120	0.0121	0.0121	0.0121	0.0144	0.0143	0.0143
Stackex-chess	0.1	0.0099	0.0099	0.0105	0.0097	0.0107	0.0119	0.0119	0.0118
	0.2	0.0101	0.0099	0.0103	0.0099	0.0107	0.0115	0.0114	0.0114
	0.3	0.0102	0.0099	0.0101	0.0100	0.0106	0.0110	0.0110	0.0112
	0.4	0.0104	0.0102	0.0102	0.0103	0.0106	0.0106	0.0108	0.0108
	0.5	0.0106	0.0103	0.0105	0.0105	0.0106	0.0106	0.0106	0.0105
	0.6	0.0106	0.0104	0.0106	0.0105	0.0106	0.0104	0.0104	0.0104
stackex-cooking	0.1	0.0050	0.0050	0.0053	0.0052	0.0057	0.0058	0.0058	0.0058
	0.2	0.0051	0.0050	0.0051	0.0054	0.0056	0.0057	0.0057	0.0057
	0.3	0.0052	0.0051	0.0051	0.0056	0.0056	0.0056	0.0056	0.0055
	0.4	0.0053	0.0052	0.0052	0.0057	0.0056	0.0055	0.0056	0.0055
	0.5	0.0054	0.0053	0.0053	0.0059	0.0056	0.0055	0.0055	0.0054
	0.6	0.0055	0.0054	0.0054	0.0056	0.0056	0.0055	0.0056	0.0056
stackex-cs	0.1	0.0088	0.0088	0.0093	0.0087	0.0094	0.0109	0.0107	0.0107
	0.2	0.0090	0.0088	0.0090	0.0088	0.0094	0.0106	0.0107	0.0107
	0.3	0.0091	0.0089	0.0090	0.0090	0.0093	0.0103	0.0103	0.0103
	0.4	0.0093	0.0090	0.0091	0.0096	0.0094	0.0102	0.0102	0.0101
	0.5	0.0093	0.0091	0.0092	0.0098	0.0094	0.0100	0.0101	0.0100
	0.6	0.0093	0.0092	0.0093	0.0094	0.0094	0.0097	0.0097	0.0096
stackex-philosophy	0.1	0.0092	0.0093	0.0090	0.0089	0.0098	0.0119	0.0118	0.0118
	0.2	0.0093	0.0093	0.0090	0.0091	0.0098	0.0115	0.0115	0.0116
	0.3	0.0095	0.0094	0.0091	0.0091	0.0098	0.0111	0.0110	0.0111
	0.4	0.0096	0.0094	0.0093	0.0094	0.0098	0.0106	0.0107	0.0106
	0.5	0.0097	0.0095	0.0095	0.0095	0.0094	0.0098	0.0102	0.0103
	0.6	0.0097	0.0096	0.0096	0.0096	0.0094	0.0098	0.0099	0.0098
rcv1subest1	0.1	0.0265	0.0276	0.0295	0.0263	0.0286	0.0290	0.0291	0.0290
	0.2	0.0267	0.0275	0.0293	0.0267	0.0285	0.0293	0.0291	0.0292
	0.3	0.0274	0.0275	0.0292	0.0270	0.0285	0.0301	0.0300	0.0301
	0.4	0.0279	0.0274	0.0289	0.0277	0.0285	0.0295	0.0295	0.0295
	0.5	0.0284	0.0278	0.0286	0.0283	0.0285	0.0288	0.0290	0.0289
	0.6	0.0285	0.0282	0.0286	0.0283	0.0286	0.0287	0.0285	0.0286
rcv1subest2	0.1	0.0247	0.0249	0.0246	0.0237	0.0261	0.0258	0.0256	0.0257
	0.2	0.0253	0.0252	0.0255	0.0240	0.0261	0.0259	0.0261	0.0260
	0.3	0.0257	0.0258	0.0255	0.0249	0.0261	0.0270	0.0274	0.0268
	0.4	0.0256	0.0256	0.0253	0.0250	0.0260	0.0270	0.0269	0.0269
	0.5	0.0256	0.0255	0.0253	0.0251	0.0261	0.0257	0.0255	0.0259
	0.6	0.0257	0.0256	0.0253	0.0251	0.0260	0.0255	0.0255	0.0253
rcv1subest3	0.1	0.0243	0.0247	0.0245	0.0235	0.0259	0.0258	0.0258	0.0258
	0.2	0.0250	0.0249	0.0251	0.0238	0.0259	0.0257	0.0255	0.0255
	0.3	0.0251	0.0252	0.0251	0.0245	0.0259	0.0266	0.0266	0.0266
	0.4	0.0251	0.0252	0.0249	0.0246	0.0260	0.0263	0.0268	0.0264
	0.5	0.0252	0.0252	0.0250	0.0247	0.0258	0.0257	0.0256	0.0259
	0.6	0.0253	0.0253	0.0250	0.0248	0.0258	0.0251	0.0246	0.0253

Table 3 Experimental results of each comparing algorithm on eleven data sets in terms of SA. For SA, the bigger the better. Best results are highlighted in bold

SA	Missing rate	MLKNN	LPLC	LIFT	LLSF	Glocal	LSML	LSLC	LSLCnew
cal500	0.1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
medical	0.1	0.4539	0.4753	0.3256	0.5264	0.0000	0.6495	0.6517	0.6500
	0.2	0.3921	0.4230	0.2698	0.4918	0.0003	0.6373	0.6219	0.6446
	0.3	0.3495	0.3823	0.2235	0.4189	0.0007	0.6014	0.5889	0.6010
	0.4	0.2555	0.3107	0.1958	0.3455	0.0008	0.5794	0.5820	0.5847
	0.5	0.1434	0.2220	0.1495	0.2471	0.0014	0.5608	0.5579	0.5550
	0.6	0.0381	0.1177	0.0351	0.1124	0.0025	0.4881	0.5235	0.5074
genbase	0.1	0.8695	0.9154	0.7373	0.8842	0.0000	0.9535	0.9527	0.9505
	0.2	0.8274	0.8490	0.7583	0.8222	0.0002	0.8978	0.8975	0.8849
	0.3	0.8039	0.7655	0.7395	0.7924	0.0003	0.8582	0.8479	0.8631
	0.4	0.4606	0.4272	0.7040	0.7225	0.0009	0.8575	0.8622	0.8614
	0.5	0.2790	0.2684	0.5930	0.6344	0.0012	0.8459	0.8469	0.8404
	0.6	0.1732	0.1820	0.1971	0.4015	0.0011	0.8383	0.8393	0.8290
stackex-chemistry	0.1	0.0299	0.0445	0.0323	0.0432	0.0000	0.0475	0.0439	0.0469
	0.2	0.0232	0.0369	0.0208	0.0336	0.0000	0.0479	0.0494	0.0446
	0.3	0.0133	0.0287	0.0141	0.0228	0.0000	0.0517	0.0483	0.0519
	0.4	0.0079	0.0219	0.0063	0.0120	0.0000	0.0529	0.0510	0.0514
	0.5	0.0049	0.0159	0.0055	0.0051	0.0000	0.0570	0.0531	0.0544
	0.6	0.0030	0.0114	0.0021	0.0040	0.0000	0.0603	0.0598	0.0600
Stackex-chess	0.1	0.0402	0.0504	0.0284	0.0580	0.0000	0.0514	0.0523	0.0512
	0.2	0.0334	0.0416	0.0303	0.0459	0.0000	0.0577	0.0584	0.0597
	0.3	0.0253	0.0383	0.0307	0.0402	0.0003	0.0597	0.0621	0.0609
	0.4	0.0142	0.0275	0.0235	0.0248	0.0005	0.0664	0.0667	0.0647
	0.5	0.0036	0.0179	0.0089	0.0131	0.0005	0.0625	0.0615	0.0634
	0.6	0.0036	0.0123	0.0024	0.0116	0.0008	0.0622	0.0601	0.0616
stackex-cooking	0.1	0.0711	0.0778	0.0773	0.0726	0.0000	0.0893	0.0875	0.0895
	0.2	0.0628	0.0715	0.0790	0.0599	0.0000	0.0934	0.0917	0.0918
	0.3	0.0520	0.0641	0.0688	0.0456	0.0000	0.0952	0.0943	0.0985
	0.4	0.0383	0.0549	0.0562	0.0310	0.0000	0.0977	0.0958	0.0943
	0.5	0.0273	0.0450	0.0418	0.0210	0.0000	0.0958	0.0945	0.0982
	0.6	0.0190	0.0346	0.0269	0.0175	0.0000	0.0947	0.0901	0.0929
stackex-cs	0.1	0.0307	0.0442	0.0305	0.0441	0.0000	0.0396	0.0423	0.0414
	0.2	0.0202	0.0369	0.0322	0.0322	0.0000	0.0444	0.0424	0.0417
	0.3	0.0141	0.0293	0.0262	0.0216	0.0000	0.0462	0.0469	0.0475
	0.4	0.0097	0.0220	0.0183	0.0121	0.0000	0.0489	0.0471	0.0502
	0.5	0.0069	0.0147	0.0071	0.0073	0.0000	0.0505	0.0477	0.0485
	0.6	0.0057	0.0105	0.0043	0.0067	0.0000	0.0508	0.0501	0.0520
stackex-philosophy	0.1	0.0612	0.0701	0.0729	0.0880	0.0000	0.0676	0.0678	0.0709
	0.2	0.0491	0.0637	0.0631	0.0767	0.0000	0.0707	0.0719	0.0699
	0.3	0.0369	0.0517	0.0519	0.0646	0.0000	0.0760	0.0803	0.0759
	0.4	0.0237	0.0431	0.0417	0.0447	0.0000	0.0812	0.0817	0.0820
	0.5	0.0140	0.0309	0.0253	0.0311	0.0000	0.0001	0.0875	0.0870
	0.6	0.0072	0.0221	0.0079	0.0265	0.0000	0.0000	0.0951	0.0948
rcv1subest1	0.1	0.0457	0.1315	0.0116	0.0598	0.0000	0.0463	0.0453	0.0445
	0.2	0.0409	0.1482	0.0140	0.0580	0.0000	0.0639	0.0655	0.0631
	0.3	0.0333	0.1622	0.0163	0.0548	0.0000	0.0974	0.0931	0.0973
	0.4	0.0070	0.1146	0.0063	0.0294	0.0000	0.0868	0.0857	0.0852
	0.5	0.0052	0.0593	0.0005	0.0232	0.0000	0.0639	0.0619	0.0636
	0.6	0.0085	0.0385	0.0002	0.0203	0.0000	0.0341	0.0347	0.0345
rcv1subest2	0.1	0.1180	0.1981	0.0789	0.1643	0.0000	0.1499	0.1533	0.1519
	0.2	0.0971	0.1730	0.0537	0.1509	0.0000	0.1590	0.1551	0.1577
	0.3	0.0951	0.1670	0.0634	0.1301	0.0000	0.1664	0.1632	0.1720
	0.4	0.0826	0.1396	0.0730	0.1192	0.0000	0.1578	0.1593	0.1580
	0.5	0.0746	0.1172	0.0724	0.1033	0.0000	0.1343	0.1357	0.1369
	0.6	0.0722	0.1102	0.0719	0.0987	0.0000	0.1108	0.1121	0.1109
rcv1subest3	0.1	0.1209	0.2006	0.0867	0.1690	0.0000	0.1518	0.1516	0.1479
	0.2	0.1007	0.1781	0.0736	0.1524	0.0000	0.1637	0.1666	0.1673
	0.3	0.1022	0.1722	0.0711	0.1325	0.0000	0.1703	0.1727	0.1764
	0.4	0.0897	0.1523	0.0822	0.1258	0.0000	0.1668	0.1647	0.1667
	0.5	0.0857	0.1271	0.0817	0.1123	0.0000	0.1389	0.1434	0.1396
	0.6	0.0814	0.1161	0.0800	0.1019	0.0000	0.1182	0.1077	0.1139

Table 4 Experimental results of each comparing algorithm on eleven data sets in terms of MF1. For MF1, the bigger the better. Best results are highlighted in bold

MF1	Missing rate	MLKNN	LPLC	LIFT	LLSF	Glocal	LSML	LSLC	LSLCnew
cal500	0.1	0.2644	0.4054	0.4584	0.2580	0.0000	0.4790	0.4756	0.4785
	0.2	0.1925	0.3699	0.4569	0.1670	0.0000	0.4781	0.4787	0.4776
	0.3	0.1091	0.3252	0.4494	0.0879	0.0000	0.4723	0.4708	0.4699
	0.4	0.0601	0.2855	0.4566	0.0559	0.0000	0.4698	0.4715	0.4697
	0.5	0.0171	0.2189	0.4341	0.0057	0.0000	0.4620	0.4609	0.4616
	0.6	0.0035	0.1475	0.2417	0.0017	0.0000	0.4535	0.4524	0.4493
medical	0.1	0.6263	0.6314	0.5207	0.7022	0.0000	0.8083	0.8111	0.8156
	0.2	0.5502	0.5698	0.4559	0.6580	0.0005	0.7907	0.7852	0.7958
	0.3	0.4852	0.5162	0.3924	0.5741	0.0011	0.7618	0.7514	0.7542
	0.4	0.3801	0.4370	0.3534	0.4948	0.0013	0.7357	0.7404	0.7383
	0.5	0.2345	0.3386	0.2862	0.3725	0.0023	0.7174	0.7147	0.7115
	0.6	0.0687	0.1919	0.0732	0.1904	0.0039	0.6520	0.6794	0.6663
genbase	0.1	0.9115	0.9526	0.8038	0.9257	0.0009	0.9780	0.9756	0.9755
	0.2	0.8395	0.8768	0.7949	0.8419	0.0024	0.9396	0.9403	0.9349
	0.3	0.8049	0.7856	0.7695	0.7965	0.0017	0.8943	0.8861	0.8973
	0.4	0.5679	0.5346	0.7393	0.7441	0.0033	0.8880	0.8933	0.8932
	0.5	0.3828	0.3636	0.6576	0.6767	0.0035	0.8747	0.8753	0.8714
	0.6	0.2515	0.2572	0.2553	0.4907	0.0046	0.8699	0.8660	0.8619
stackex-chemistry	0.1	0.1054	0.1631	0.1261	0.1652	0.0000	0.3639	0.3609	0.3617
	0.2	0.0788	0.1324	0.0845	0.1261	0.0000	0.3591	0.3584	0.3587
	0.3	0.0472	0.0998	0.0542	0.0867	0.0000	0.3552	0.3536	0.3559
	0.4	0.0249	0.0724	0.0255	0.0452	0.0000	0.3466	0.3441	0.3450
	0.5	0.0143	0.0497	0.0169	0.0158	0.0000	0.3311	0.3349	0.3337
	0.6	0.0070	0.0320	0.0048	0.0099	0.0000	0.3186	0.3208	0.3168
Stackex-chess	0.1	0.2096	0.2606	0.1764	0.2873	0.0000	0.4216	0.4217	0.4212
	0.2	0.1692	0.2305	0.1581	0.2353	0.0000	0.4112	0.4135	0.4125
	0.3	0.1148	0.1871	0.1443	0.1901	0.0003	0.4025	0.4033	0.3962
	0.4	0.0629	0.1373	0.1068	0.1241	0.0010	0.3833	0.3817	0.3787
	0.5	0.0175	0.0842	0.0376	0.0720	0.0011	0.3528	0.3477	0.3520
	0.6	0.0082	0.0440	0.0030	0.0477	0.0016	0.3142	0.3090	0.3090
stackex-cooking	0.1	0.2527	0.2935	0.3652	0.2841	0.0000	0.4529	0.4532	0.4532
	0.2	0.2084	0.2609	0.3281	0.2232	0.0000	0.4519	0.4496	0.4532
	0.3	0.1611	0.2254	0.2593	0.1610	0.0000	0.4471	0.4477	0.4480
	0.4	0.1011	0.1802	0.1877	0.0888	0.0000	0.4467	0.4459	0.4455
	0.5	0.0542	0.1317	0.1152	0.0385	0.0000	0.4386	0.4375	0.4399
	0.6	0.0238	0.0887	0.0495	0.0202	0.0000	0.4319	0.4326	0.4304
stackex-cs	0.1	0.1928	0.2711	0.2877	0.2710	0.0000	0.4571	0.4594	0.4565
	0.2	0.1208	0.2260	0.2585	0.1961	0.0000	0.4572	0.4539	0.4547
	0.3	0.0713	0.1769	0.2050	0.1164	0.0001	0.4514	0.4517	0.4516
	0.4	0.0311	0.1239	0.1293	0.0453	0.0001	0.4457	0.4457	0.4451
	0.5	0.0153	0.0788	0.0474	0.0228	0.0002	0.4316	0.4370	0.4356
	0.6	0.0066	0.0411	0.0035	0.0147	0.0002	0.4141	0.4161	0.4146
stackex-philosophy	0.1	0.1781	0.2150	0.2342	0.2852	0.0000	0.4030	0.4053	0.4029
	0.2	0.1388	0.1844	0.2035	0.2420	0.0000	0.3987	0.3973	0.3994
	0.3	0.1004	0.1472	0.1694	0.1979	0.0001	0.3922	0.3913	0.3914
	0.4	0.0564	0.1133	0.1321	0.1411	0.0002	0.3784	0.3758	0.3819
	0.5	0.0291	0.0764	0.0749	0.0899	0.0002	0.0005	0.3611	0.3603
	0.6	0.0107	0.0450	0.0166	0.0633	0.0002	0.0007	0.3388	0.3398
rcv1subest1	0.1	0.2634	0.4512	0.2191	0.3178	0.0000	0.5102	0.5109	0.5112
	0.2	0.2292	0.4039	0.1805	0.2737	0.0000	0.4911	0.4901	0.4934
	0.3	0.1905	0.3683	0.1561	0.2294	0.0000	0.4638	0.4656	0.4651
	0.4	0.0567	0.3049	0.0823	0.1532	0.0001	0.4549	0.4533	0.4556
	0.5	0.0153	0.1926	0.0048	0.0926	0.0001	0.4277	0.4293	0.4287
	0.6	0.0104	0.0887	0.0002	0.0583	0.0001	0.3852	0.3807	0.3804
rcv1subest2	0.1	0.2274	0.3353	0.1293	0.3282	0.0000	0.5016	0.4975	0.5002
	0.2	0.1444	0.2739	0.0639	0.2866	0.0000	0.4881	0.4839	0.4862
	0.3	0.1261	0.2418	0.0655	0.2318	0.0000	0.4556	0.4561	0.4575
	0.4	0.0848	0.1821	0.0618	0.1801	0.0000	0.4319	0.4324	0.4307
	0.5	0.0625	0.1265	0.0605	0.1339	0.0000	0.4156	0.4139	0.4207
	0.6	0.0569	0.1010	0.0591	0.1161	0.0000	0.3832	0.3816	0.3825
rcv1subest3	0.1	0.2248	0.3245	0.1211	0.3253	0.0000	0.5056	0.5055	0.5051
	0.2	0.1485	0.2688	0.0742	0.2815	0.0000	0.4836	0.4841	0.4831
	0.3	0.1372	0.2326	0.0761	0.2303	0.0000	0.4571	0.4560	0.4549
	0.4	0.0956	0.1892	0.0716	0.1877	0.0000	0.4325	0.4353	0.4349
	0.5	0.0747	0.1334	0.0676	0.1460	0.0000	0.4121	0.4146	0.4153
	0.6	0.0658	0.1037	0.0663	0.1219	0.0000	0.3826	0.0915	0.3803

Table 5 Experimental results of each comparing algorithm on eleven data sets in terms of AP. For AP, the bigger the better. Best results are highlighted in bold

AP	Missing rate	MLKNN	LPLC	LIFT	LLSF	Glocal	LSML	LSLC	LSLCnew
cal500	0.1	0.4904	0.4575	0.4706	0.4954	0.4919	0.5020	0.4984	0.5010
	0.2	0.4872	0.4451	0.4701	0.4938	0.4940	0.5012	0.5016	0.4997
	0.3	0.4804	0.4287	0.4607	0.4918	0.4971	0.4976	0.4970	0.4966
	0.4	0.4809	0.4223	0.4705	0.4859	0.4999	0.5000	0.4991	0.4998
	0.5	0.4748	0.4033	0.4610	0.4863	0.4979	0.4980	0.4963	0.4959
	0.6	0.4682	0.3807	0.3333	0.4826	0.4979	0.4958	0.4942	0.4915
medical	0.1	0.8057	0.7727	0.7566	0.8620	0.6696	0.9003	0.9015	0.9042
	0.2	0.7944	0.7648	0.7550	0.8610	0.7810	0.8947	0.8909	0.8986
	0.3	0.7906	0.7589	0.7565	0.8505	0.8374	0.8864	0.8795	0.8801
	0.4	0.7720	0.7397	0.5988	0.8300	0.8589	0.8744	0.8747	0.8752
	0.5	0.7662	0.7339	0.5376	0.8078	0.8699	0.8693	0.8682	0.8692
	0.6	0.7452	0.7035	0.4859	0.7678	0.8724	0.8473	0.8503	0.8489
genbase	0.1	0.9866	0.9889	0.9621	0.9895	0.9424	0.9929	0.9925	0.9918
	0.2	0.9790	0.9814	0.8950	0.9713	0.9847	0.9880	0.9873	0.9877
	0.3	0.9670	0.9706	0.8395	0.9584	0.9924	0.9787	0.9768	0.9802
	0.4	0.8947	0.7385	0.8276	0.9492	0.9950	0.9736	0.9767	0.9716
	0.5	0.7987	0.6059	0.7626	0.9361	0.9942	0.9678	0.9714	0.9672
	0.6	0.7568	0.5441	0.3691	0.8679	0.9950	0.9617	0.9657	0.9604
stackex-chemistry	0.1	0.3810	0.3503	0.2827	0.3843	0.2964	0.4572	0.4573	0.4563
	0.2	0.3740	0.3385	0.2480	0.3665	0.3498	0.4513	0.4501	0.4516
	0.3	0.3662	0.3273	0.1990	0.3522	0.3764	0.4465	0.4453	0.4476
	0.4	0.3562	0.3126	0.1431	0.3285	0.3848	0.4367	0.4335	0.4367
	0.5	0.3489	0.2989	0.1222	0.3023	0.3891	0.4246	0.4286	0.4243
	0.6	0.3368	0.2800	0.1135	0.2919	0.3895	0.4125	0.4127	0.4114
Stackex-chess	0.1	0.4175	0.3894	0.2925	0.4639	0.2689	0.5005	0.4990	0.4992
	0.2	0.4067	0.3749	0.2670	0.4385	0.3545	0.4825	0.4899	0.4846
	0.3	0.4010	0.3655	0.2396	0.4266	0.3950	0.4760	0.4743	0.4720
	0.4	0.3895	0.3469	0.2180	0.4000	0.4104	0.4632	0.4615	0.4582
	0.5	0.3717	0.3236	0.2142	0.3774	0.4133	0.4408	0.4408	0.4388
	0.6	0.3555	0.3035	0.1835	0.3499	0.4235	0.4198	0.4137	0.4216
stackex-cooking	0.1	0.4146	0.3912	0.4269	0.4124	0.2884	0.5144	0.5151	0.5167
	0.2	0.4072	0.3810	0.4187	0.3787	0.3115	0.5115	0.5100	0.5124
	0.3	0.3993	0.3702	0.4040	0.3349	0.3004	0.5059	0.5052	0.5076
	0.4	0.3890	0.3586	0.3782	0.2830	0.2839	0.5026	0.5023	0.5015
	0.5	0.3750	0.3381	0.3383	0.2382	0.2674	0.4933	0.4937	0.4930
	0.6	0.3592	0.3171	0.2718	0.2132	0.2570	0.4852	0.4839	0.4816
stackex-cs	0.1	0.4325	0.4037	0.4022	0.4522	0.3502	0.5245	0.5243	0.5224
	0.2	0.4254	0.3946	0.3781	0.4341	0.4030	0.5216	0.5191	0.5192
	0.3	0.4179	0.3814	0.3554	0.3959	0.4167	0.5135	0.5144	0.5141
	0.4	0.4067	0.3662	0.3161	0.3383	0.4187	0.5066	0.5066	0.5072
	0.5	0.3924	0.3436	0.2454	0.3022	0.4194	0.4940	0.4958	0.4958
	0.6	0.3741	0.3165	0.1851	0.2892	0.4174	0.4780	0.4797	0.4794
stackex-philosophy	0.1	0.4000	0.3676	0.3549	0.4676	0.3164	0.4965	0.4990	0.4971
	0.2	0.3944	0.3581	0.3311	0.4526	0.3794	0.4894	0.4877	0.4895
	0.3	0.3863	0.3453	0.3041	0.4440	0.4071	0.4799	0.4802	0.4804
	0.4	0.3749	0.3313	0.2840	0.4185	0.4156	0.4657	0.4637	0.4691
	0.5	0.3640	0.3113	0.2634	0.3906	0.4194	0.4227	0.4503	0.4502
	0.6	0.3549	0.2961	0.2390	0.3686	0.4174	0.4235	0.4357	0.4386
rcv1subest1	0.1	0.6163	0.6010	0.4097	0.5979	0.4575	0.6119	0.6099	0.6115
	0.2	0.6083	0.5859	0.3758	0.5834	0.5335	0.6019	0.6025	0.6033
	0.3	0.5885	0.5680	0.3543	0.5655	0.5612	0.5809	0.5800	0.5792
	0.4	0.5796	0.5569	0.2864	0.5401	0.5774	0.5721	0.5695	0.5736
	0.5	0.5645	0.5359	0.2036	0.5130	0.5813	0.5583	0.5596	0.5588
	0.6	0.5426	0.5013	0.1535	0.4847	0.5849	0.5359	0.5345	0.5318
rcv1subest2	0.1	0.6053	0.5944	0.4222	0.6100	0.4560	0.6366	0.6357	0.6360
	0.2	0.5988	0.5847	0.3967	0.5990	0.5308	0.6289	0.6252	0.6255
	0.3	0.5808	0.5636	0.3607	0.5775	0.5663	0.6072	0.6062	0.6117
	0.4	0.5678	0.5454	0.3241	0.5550	0.5794	0.5926	0.5950	0.5948
	0.5	0.5548	0.5265	0.2977	0.5316	0.5855	0.5838	0.5845	0.5854
	0.6	0.5311	0.4975	0.2896	0.5130	0.5892	0.5553	0.5552	0.5577
rcv1subest3	0.1	0.6080	0.5970	0.4351	0.6057	0.4577	0.6310	0.6298	0.6293
	0.2	0.5983	0.5833	0.4046	0.5945	0.5332	0.6226	0.6262	0.6234
	0.3	0.5862	0.5673	0.3699	0.5808	0.5657	0.6088	0.6076	0.6079
	0.4	0.5698	0.5473	0.3333	0.5600	0.5758	0.5930	0.5964	0.5961
	0.5	0.5567	0.5297	0.2865	0.5345	0.5810	0.5792	0.5828	0.5803
	0.6	0.5340	0.5003	0.2855	0.5152	0.5887	0.5551	0.5568	0.5525

Table 6 Experimental results of each comparing algorithm on eleven data sets in terms of OE. For OE, the smaller the better. Best results are highlighted in bold

OE	Missing rate	MLKNN	LPLC	LIFT	LLSF	Glocal	LSML	LSLC	LSLCnew
cal500	0.1	0.1248	0.2131	0.3263	0.1344	0.1163	0.1136	0.1202	0.1123
	0.2	0.1485	0.2234	0.2974	0.1427	0.1153	0.1207	0.1141	0.1206
	0.3	0.1757	0.2657	0.3149	0.1292	0.1169	0.1235	0.1149	0.1159
	0.4	0.1958	0.2914	0.2029	0.2165	0.1136	0.1079	0.1164	0.1131
	0.5	0.2207	0.3188	0.2025	0.1813	0.1206	0.1187	0.1196	0.1137
	0.6	0.2477	0.3443	0.1572	0.1839	0.1145	0.1148	0.1272	0.1223
medical	0.1	0.2583	0.2865	0.2506	0.1780	0.4302	0.1382	0.1368	0.1317
	0.2	0.2689	0.2878	0.2457	0.1738	0.2815	0.1455	0.1503	0.1435
	0.3	0.2714	0.2831	0.2423	0.1831	0.2091	0.1500	0.1581	0.1573
	0.4	0.2899	0.2933	0.3818	0.2033	0.1801	0.1633	0.1671	0.1630
	0.5	0.2958	0.2825	0.4319	0.2349	0.1663	0.1667	0.1700	0.1724
	0.6	0.3195	0.2810	0.4882	0.2865	0.1639	0.1993	0.1918	0.2004
genbase	0.1	0.0080	0.0026	0.0059	0.0018	0.0775	0.0027	0.0024	0.0027
	0.2	0.0136	0.0054	0.0611	0.0060	0.0147	0.0032	0.0048	0.0021
	0.3	0.0136	0.0042	0.1264	0.0071	0.0048	0.0032	0.0036	0.0027
	0.4	0.1419	0.2620	0.1413	0.0133	0.0023	0.0062	0.0054	0.0103
	0.5	0.2778	0.4077	0.2274	0.0252	0.0027	0.0107	0.0094	0.0118
	0.6	0.3405	0.4732	0.4798	0.1409	0.0032	0.0153	0.0127	0.0136
stackex-chemistry	0.1	0.6243	0.5658	0.6715	0.6191	0.7271	0.5479	0.5482	0.5496
	0.2	0.6332	0.5683	0.7012	0.6382	0.6648	0.5541	0.5574	0.5530
	0.3	0.6434	0.5706	0.7492	0.6523	0.6317	0.5589	0.5592	0.5543
	0.4	0.6516	0.5743	0.8144	0.6886	0.6179	0.5637	0.5724	0.5659
	0.5	0.6593	0.5700	0.8413	0.7236	0.6092	0.5746	0.5695	0.5769
	0.6	0.6695	0.5721	0.8576	0.7329	0.6112	0.5826	0.5824	0.5837
Stackex-chess	0.1	0.5077	0.4927	0.5653	0.4619	0.6869	0.4268	0.4339	0.4240
	0.2	0.5227	0.4983	0.5893	0.4884	0.5810	0.4399	0.4376	0.4401
	0.3	0.5271	0.5011	0.6243	0.5067	0.5305	0.4515	0.4544	0.4572
	0.4	0.5398	0.5172	0.6386	0.5405	0.5033	0.4612	0.4634	0.4660
	0.5	0.5562	0.5257	0.6438	0.5699	0.5074	0.4793	0.4850	0.4841
	0.6	0.5767	0.5264	0.6930	0.6039	0.4922	0.5071	0.5189	0.5077
stackex-cooking	0.1	0.4796	0.4344	0.4585	0.5170	0.6791	0.4121	0.4134	0.4139
	0.2	0.4886	0.4382	0.4611	0.5572	0.6297	0.4131	0.4157	0.4147
	0.3	0.4945	0.4373	0.4732	0.6241	0.6406	0.4169	0.4190	0.4152
	0.4	0.5067	0.4392	0.4882	0.7079	0.6575	0.4202	0.4190	0.4210
	0.5	0.5238	0.4469	0.5178	0.7634	0.6760	0.4265	0.4266	0.4259
	0.6	0.5430	0.4531	0.5825	0.7953	0.6925	0.4297	0.4312	0.4317
stackex-cs	0.1	0.5255	0.4755	0.5494	0.4923	0.6084	0.4404	0.4402	0.4430
	0.2	0.5327	0.4777	0.5668	0.4969	0.5459	0.4422	0.4458	0.4453
	0.3	0.5389	0.4775	0.5762	0.5391	0.5268	0.4478	0.4449	0.4465
	0.4	0.5493	0.4779	0.5926	0.6218	0.5229	0.4531	0.4508	0.4502
	0.5	0.5608	0.4838	0.6429	0.6892	0.5179	0.4630	0.4603	0.4591
	0.6	0.5807	0.4908	0.7068	0.6880	0.5233	0.4762	0.4709	0.4720
stackex-philosophy	0.1	0.5645	0.5590	0.5283	0.4818	0.6775	0.4531	0.4489	0.4530
	0.2	0.5753	0.5637	0.5488	0.4960	0.5898	0.4576	0.4615	0.4553
	0.3	0.5796	0.5678	0.5708	0.5017	0.5512	0.4661	0.4650	0.4603
	0.4	0.5911	0.5718	0.5956	0.5388	0.5387	0.4780	0.4808	0.4761
	0.5	0.6065	0.5710	0.6222	0.5771	0.5179	0.5273	0.4949	0.4937
	0.6	0.6132	0.5574	0.6558	0.6038	0.5233	0.5289	0.5089	0.5053
rcv1subest1	0.1	0.4226	0.4240	0.5852	0.4225	0.5525	0.4212	0.4279	0.4218
	0.2	0.4341	0.4399	0.6268	0.4406	0.4832	0.4351	0.4334	0.4329
	0.3	0.4525	0.4481	0.6431	0.4541	0.4581	0.4592	0.4602	0.4677
	0.4	0.4617	0.4562	0.7181	0.4881	0.4442	0.4658	0.4681	0.4624
	0.5	0.4758	0.4613	0.7973	0.5233	0.4394	0.4753	0.4672	0.4711
	0.6	0.4875	0.4595	0.8680	0.5547	0.4379	0.4834	0.4901	0.4890
rcv1subest2	0.1	0.4829	0.4096	0.5836	0.4454	0.5601	0.4072	0.4104	0.4086
	0.2	0.4924	0.4211	0.5973	0.4561	0.4910	0.4175	0.4242	0.4240
	0.3	0.5138	0.4483	0.6489	0.4745	0.4592	0.4396	0.4429	0.4368
	0.4	0.5233	0.4437	0.6715	0.4952	0.4456	0.4505	0.4463	0.4474
	0.5	0.5246	0.4345	0.6757	0.5103	0.4362	0.4478	0.4484	0.4456
	0.6	0.5347	0.4307	0.6829	0.5229	0.4328	0.4644	0.4654	0.4629
rcv1subest3	0.1	0.4762	0.4060	0.5701	0.4533	0.5649	0.4161	0.4200	0.4197
	0.2	0.4957	0.4246	0.5866	0.4655	0.4879	0.4289	0.4231	0.4250
	0.3	0.5108	0.4389	0.6291	0.4733	0.4576	0.4411	0.4448	0.4417
	0.4	0.5208	0.4443	0.6462	0.4921	0.4429	0.4541	0.4463	0.4491
	0.5	0.5234	0.4310	0.6877	0.5140	0.4407	0.4572	0.4530	0.4527
	0.6	0.5266	0.4264	0.6832	0.5230	0.4280	0.4677	0.4638	0.4665

Table 7 Experimental results of each comparing algorithm on eleven data sets in terms of RL. For RL, the smaller the better. Best results are highlighted in bold

RL	Missing rate	MLKNN	LPLC	LIFT	LLSF	Glocal	LSML	LSLC	LSLCnew
cal500	0.1	0.1841	0.2310	0.1993	0.1832	0.1878	0.1794	0.1804	0.1796
	0.2	0.1851	0.2416	0.2017	0.1838	0.1856	0.1798	0.1800	0.1798
	0.3	0.1873	0.2539	0.2106	0.1855	0.1822	0.1816	0.1819	0.1810
	0.4	0.1866	0.2600	0.2060	0.1858	0.1805	0.1807	0.1800	0.1812
	0.5	0.1878	0.2764	0.2130	0.1869	0.1799	0.1814	0.1833	0.1834
	0.6	0.1908	0.2981	0.5462	0.1889	0.1802	0.1833	0.1841	0.1852
medical	0.1	0.0412	0.0598	0.1409	0.0293	0.0732	0.0178	0.0171	0.0171
	0.2	0.0436	0.0646	0.1460	0.0316	0.0476	0.0198	0.0207	0.0181
	0.3	0.0478	0.0711	0.1359	0.0370	0.0339	0.0251	0.0273	0.0256
	0.4	0.0559	0.0863	0.3227	0.0428	0.0292	0.0301	0.0279	0.0294
	0.5	0.0561	0.0936	0.4161	0.0459	0.0275	0.0309	0.0318	0.0289
	0.6	0.0644	0.1193	0.5050	0.0527	0.0260	0.0367	0.0367	0.0353
genbase	0.1	0.0080	0.0061	0.0235	0.0066	0.0126	0.0051	0.0057	0.0062
	0.2	0.0123	0.0097	0.0875	0.0140	0.0057	0.0084	0.0082	0.0080
	0.3	0.0243	0.0183	0.1393	0.0246	0.0033	0.0148	0.0159	0.0138
	0.4	0.0347	0.1557	0.1541	0.0289	0.0027	0.0182	0.0163	0.0210
	0.5	0.0531	0.2333	0.2305	0.0355	0.0031	0.0227	0.0190	0.0235
	0.6	0.0632	0.2689	0.6441	0.0529	0.0020	0.0266	0.0239	0.0281
stackex-chemistry	0.1	0.1242	0.2294	0.5054	0.1174	0.1734	0.1168	0.1160	0.1173
	0.2	0.1282	0.2403	0.5601	0.1245	0.1445	0.1231	0.1234	0.1239
	0.3	0.1306	0.2530	0.6468	0.1294	0.1293	0.1294	0.1294	0.1293
	0.4	0.1356	0.2695	0.7283	0.1362	0.1236	0.1367	0.1394	0.1389
	0.5	0.1388	0.2839	0.7787	0.1439	0.1194	0.1482	0.1483	0.1478
	0.6	0.1452	0.3046	0.8037	0.1515	0.1173	0.1596	0.1619	0.1615
Stackex-chess	0.1	0.1231	0.2117	0.4319	0.1067	0.2253	0.1138	0.1129	0.1151
	0.2	0.1285	0.2244	0.4519	0.1167	0.1672	0.1251	0.1247	0.1248
	0.3	0.1327	0.2358	0.4912	0.1220	0.1445	0.1325	0.1311	0.1285
	0.4	0.1366	0.2488	0.6574	0.1301	0.1338	0.1428	0.1444	0.1437
	0.5	0.1458	0.2697	0.7144	0.1425	0.1257	0.1585	0.1575	0.1553
	0.6	0.1534	0.2900	0.7654	0.1585	0.1200	0.1742	0.1735	0.1684
stackex-cooking	0.1	0.1367	0.2262	0.3101	0.1210	0.1804	0.0952	0.0945	0.0938
	0.2	0.1401	0.2344	0.3266	0.1286	0.1575	0.0986	0.0988	0.0981
	0.3	0.1449	0.2445	0.3420	0.1387	0.1527	0.1039	0.1036	0.1034
	0.4	0.1506	0.2536	0.3837	0.1500	0.1563	0.1088	0.1089	0.1079
	0.5	0.1569	0.2699	0.4449	0.1603	0.1631	0.1150	0.1152	0.1143
	0.6	0.1643	0.2855	0.5554	0.1670	0.1654	0.1228	0.1235	0.1243
stackex-cs	0.1	0.1093	0.1947	0.2873	0.0930	0.1393	0.0737	0.0739	0.0741
	0.2	0.1122	0.2040	0.3305	0.0983	0.1110	0.0770	0.0773	0.0774
	0.3	0.1160	0.2142	0.3740	0.1076	0.1015	0.0824	0.0824	0.0817
	0.4	0.1204	0.2274	0.4545	0.1175	0.0996	0.0879	0.0878	0.0887
	0.5	0.1269	0.2448	0.5858	0.1251	0.0978	0.0959	0.0952	0.0959
	0.6	0.1330	0.2671	0.6835	0.1296	0.0971	0.1056	0.1056	0.1049
stackex-philosophy	0.1	0.1254	0.2203	0.4284	0.0960	0.1806	0.1204	0.1209	0.1194
	0.2	0.1259	0.2291	0.4655	0.1003	0.1488	0.1249	0.1248	0.1261
	0.3	0.1292	0.2437	0.5203	0.1044	0.1341	0.1320	0.1334	0.1342
	0.4	0.1340	0.2572	0.5656	0.1127	0.1269	0.1423	0.1437	0.1416
	0.5	0.1387	0.2763	0.6101	0.1193	0.0978	0.1206	0.1557	0.1546
	0.6	0.1441	0.2944	0.6638	0.1308	0.0971	0.1182	0.1703	0.1683
rcv1subest1	0.1	0.0539	0.0816	0.2177	0.0505	0.1035	0.0530	0.0533	0.0534
	0.2	0.0588	0.0934	0.2740	0.0556	0.0751	0.0597	0.0599	0.0601
	0.3	0.0700	0.1150	0.3054	0.0646	0.0670	0.0752	0.0752	0.0743
	0.4	0.0741	0.1241	0.3914	0.0708	0.0612	0.0805	0.0808	0.0806
	0.5	0.0796	0.1374	0.5536	0.0757	0.0602	0.0859	0.0853	0.0866
	0.6	0.0871	0.1628	0.8036	0.0805	0.0582	0.0979	0.0977	0.0990
rcv1subest2	0.1	0.0567	0.0908	0.2891	0.0512	0.1076	0.0525	0.0527	0.0532
	0.2	0.0619	0.1003	0.3626	0.0556	0.0792	0.0585	0.0593	0.0583
	0.3	0.0723	0.1202	0.3835	0.0646	0.0688	0.0717	0.0719	0.0696
	0.4	0.0792	0.1387	0.4397	0.0709	0.0650	0.0819	0.0814	0.0807
	0.5	0.0828	0.1495	0.5219	0.0747	0.0638	0.0852	0.0841	0.0844
	0.6	0.0915	0.1750	0.5719	0.0803	0.0616	0.1012	0.1012	0.1002
rcv1subest3	0.1	0.0571	0.0918	0.2818	0.0519	0.1104	0.0528	0.0519	0.0527
	0.2	0.0619	0.1013	0.3537	0.0572	0.0813	0.0585	0.0579	0.0593
	0.3	0.0714	0.1197	0.3776	0.0644	0.0693	0.0713	0.0717	0.0718
	0.4	0.0808	0.1386	0.4258	0.0709	0.0656	0.0812	0.0808	0.0804
	0.5	0.0852	0.1519	0.4862	0.0738	0.0638	0.0861	0.0849	0.0862
	0.6	0.0935	0.1761	0.5354	0.0787	0.0623	0.0989	0.0975	0.1017

Table 8 Experimental results of each comparing algorithm on eleven data sets in terms of AUC. For AUC, the bigger the better. Best results are highlighted in bold

AUC	Missing rate	MLKNN	LPLC	LIFT	LLSF	Glocal	LSML	LSLC	LSLCnew
cal500	0.1	0.8121	0.7722	0.7984	0.8131	0.8083	0.8170	0.8160	0.8167
	0.2	0.8113	0.7630	0.7961	0.8126	0.8107	0.8166	0.8164	0.8169
	0.3	0.8090	0.7516	0.7874	0.8108	0.8143	0.8148	0.8147	0.8158
	0.4	0.8101	0.7466	0.7919	0.8110	0.8159	0.8161	0.8165	0.8154
	0.5	0.8087	0.7311	0.7854	0.8098	0.8167	0.8154	0.8130	0.8132
	0.6	0.8057	0.7109	0.7201	0.8078	0.8163	0.8133	0.8123	0.8115
medical	0.1	0.9503	0.9327	0.8544	0.9622	0.9194	0.9762	0.9771	0.9769
	0.2	0.9473	0.9266	0.8481	0.9602	0.9444	0.9739	0.9723	0.9764
	0.3	0.9420	0.9175	0.8597	0.9536	0.9590	0.9682	0.9660	0.9675
	0.4	0.9334	0.9027	0.6946	0.9473	0.9633	0.9628	0.9646	0.9629
	0.5	0.9332	0.8947	0.6193	0.9440	0.9653	0.9614	0.9612	0.9635
	0.6	0.9257	0.8683	0.6103	0.9372	0.9672	0.9551	0.9568	0.9577
genbase	0.1	0.9795	0.9828	0.9521	0.9815	0.9740	0.9844	0.9832	0.9821
	0.2	0.9716	0.9750	0.8625	0.9684	0.9842	0.9800	0.9811	0.9804
	0.3	0.9503	0.9556	0.8340	0.9510	0.9878	0.9707	0.9679	0.9722
	0.4	0.9322	0.8389	0.8409	0.9426	0.9879	0.9663	0.9686	0.9636
	0.5	0.9079	0.7666	0.7884	0.9311	0.9873	0.9596	0.9630	0.9574
	0.6	0.8971	0.7326	0.7045	0.9033	0.9894	0.9551	0.9589	0.9539
stackex-chemistry	0.1	0.8665	0.7644	0.5152	0.8737	0.8173	0.8736	0.8737	0.8733
	0.2	0.8622	0.7523	0.4783	0.8667	0.8481	0.8675	0.8661	0.8671
	0.3	0.8593	0.7386	0.4294	0.8615	0.8637	0.8610	0.8603	0.8611
	0.4	0.8545	0.7225	0.4004	0.8548	0.8692	0.8528	0.8511	0.8513
	0.5	0.8505	0.7063	0.4265	0.8468	0.8738	0.8410	0.8426	0.8416
	0.6	0.8438	0.6844	0.4409	0.8392	0.8760	0.8292	0.8279	0.8284
Stackex-chess	0.1	0.8653	0.7790	0.5790	0.8852	0.7674	0.8786	0.8787	0.8776
	0.2	0.8598	0.7664	0.5605	0.8746	0.8265	0.8661	0.8660	0.8665
	0.3	0.8549	0.7554	0.5338	0.8682	0.8514	0.8592	0.8590	0.8630
	0.4	0.8505	0.7408	0.4780	0.8606	0.8606	0.8485	0.8472	0.8476
	0.5	0.8433	0.7207	0.4862	0.8491	0.8687	0.8354	0.8339	0.8361
	0.6	0.8361	0.6997	0.4785	0.8326	0.8755	0.8182	0.8199	0.8240
stackex-cooking	0.1	0.8450	0.7538	0.6698	0.8629	0.8146	0.8887	0.8896	0.8897
	0.2	0.8407	0.7449	0.6533	0.8545	0.8401	0.8846	0.8848	0.8850
	0.3	0.8356	0.7340	0.6370	0.8439	0.8456	0.8789	0.8801	0.8801
	0.4	0.8300	0.7241	0.5996	0.8328	0.8427	0.8739	0.8745	0.8755
	0.5	0.8235	0.7080	0.5508	0.8224	0.8363	0.8679	0.8670	0.8681
	0.6	0.8162	0.6918	0.4898	0.8160	0.8336	0.8599	0.8596	0.8583
stackex-cs	0.1	0.8862	0.8017	0.7095	0.9023	0.8610	0.9201	0.9191	0.9197
	0.2	0.8831	0.7919	0.6719	0.8963	0.8891	0.9162	0.9158	0.9157
	0.3	0.8786	0.7808	0.6334	0.8866	0.8987	0.9105	0.9112	0.9116
	0.4	0.8740	0.7661	0.5803	0.8764	0.9008	0.9045	0.9048	0.9036
	0.5	0.8673	0.7490	0.4924	0.8685	0.9028	0.8963	0.8966	0.8966
	0.6	0.8609	0.7257	0.4606	0.8642	0.9037	0.8863	0.8858	0.8866
stackex-philosophy	0.1	0.8567	0.7587	0.5664	0.8906	0.8022	0.8648	0.8635	0.8654
	0.2	0.8555	0.7497	0.5454	0.8863	0.8358	0.8594	0.8600	0.8594
	0.3	0.8519	0.7347	0.5165	0.8804	0.8525	0.8517	0.8498	0.8499
	0.4	0.8463	0.7203	0.4998	0.8717	0.8600	0.8411	0.8391	0.8422
	0.5	0.8415	0.7024	0.4814	0.8639	0.9028	0.8667	0.8270	0.8284
	0.6	0.8358	0.6833	0.4868	0.8516	0.9037	0.8698	0.8110	0.8128
rcv1subest1	0.1	0.9278	0.8947	0.7586	0.9338	0.8745	0.9293	0.9294	0.9290
	0.2	0.9225	0.8814	0.7014	0.9276	0.9056	0.9217	0.9216	0.9212
	0.3	0.9105	0.8582	0.6636	0.9181	0.9146	0.9054	0.9051	0.9064
	0.4	0.9060	0.8486	0.5899	0.9110	0.9206	0.8991	0.8990	0.8992
	0.5	0.9006	0.8342	0.4726	0.9051	0.9218	0.8935	0.8938	0.8923
	0.6	0.8923	0.8094	0.4721	0.9005	0.9238	0.8802	0.8802	0.8791
rcv1subest2	0.1	0.9194	0.8849	0.6661	0.9270	0.8642	0.9243	0.9244	0.9235
	0.2	0.9137	0.8740	0.5984	0.9221	0.8946	0.9173	0.9162	0.9177
	0.3	0.9019	0.8497	0.5725	0.9109	0.9066	0.9020	0.9011	0.9038
	0.4	0.8943	0.8295	0.5055	0.9038	0.9104	0.8898	0.8903	0.8910
	0.5	0.8903	0.8175	0.4608	0.9001	0.9112	0.8864	0.8872	0.8867
	0.6	0.8804	0.7890	0.4745	0.8937	0.9141	0.8694	0.8689	0.8705
rcv1subest3	0.1	0.9204	0.8840	0.6819	0.9270	0.8624	0.9258	0.9272	0.9261
	0.2	0.9148	0.8729	0.6076	0.9203	0.8941	0.9189	0.9192	0.9176
	0.3	0.9041	0.8507	0.5804	0.9116	0.9070	0.9034	0.9028	0.9026
	0.4	0.8933	0.8278	0.5312	0.9037	0.9115	0.8913	0.8912	0.8931
	0.5	0.8884	0.8124	0.4721	0.9008	0.9141	0.8861	0.8876	0.8866
	0.6	0.8794	0.7871	0.4716	0.8953	0.9154	0.8727	0.8744	0.8698

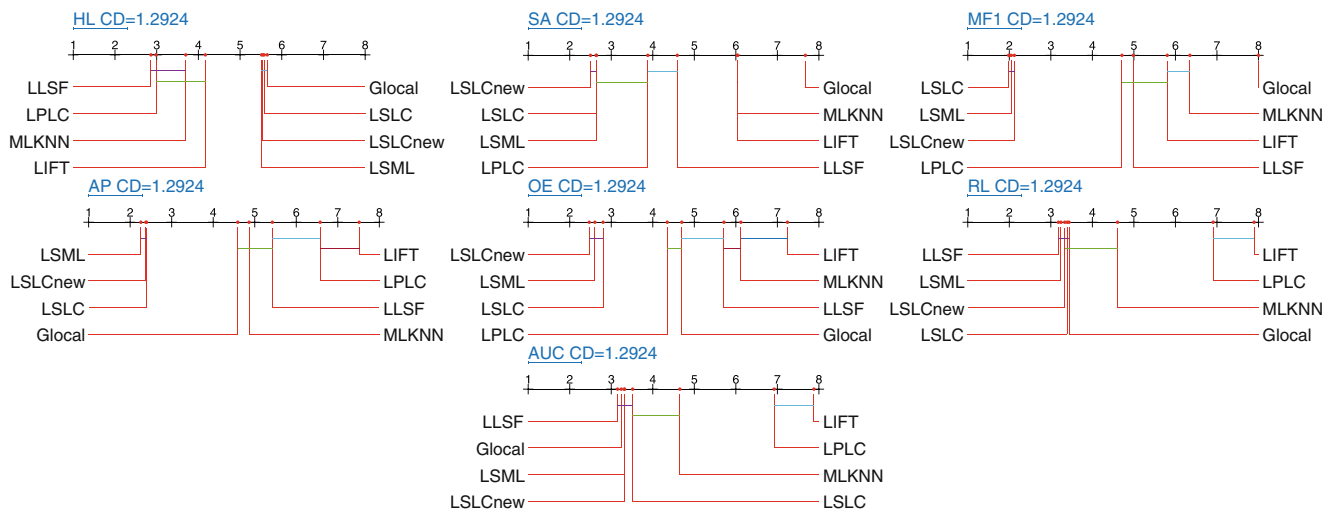


Fig. 1 The detailed friedman test diagram of the experimental results

5.4 Comparison experiments

In the comparison experiments, we control the missing rate of class labels to be in the range of 10% to 60% with a step size of 10% and all data is normalized for fairness. Thus, for every comparing algorithm, the average result(mean std) on eleven data sets in terms of evaluation metrics are consulted on Tables 2, 3, 4, 5, 6, 7, and 8. Considering the limitation of table size, the experimental results here only give the average value, and the complete experimental results(mean ± std) are recorded in the uploaded zip file containing 11 Excel files. In order to better evaluate the performance of the algorithm, Friedman test [41] and Holm test [42] are used in here. Table 10 summarizes the Friedman statistics F_F and the corresponding critical value in terms of every metric. As shown in Table 10, at significance level $\alpha = 0.05$, the full hypothesis that the compared algorithm has performance of equivalent is rejected in terms of all the evaluation metric. As a result, the Nemenyi test [41] is employed to test whether our method LSLC can perform competitiveness against comparing algorithm. The performance of two algorithms will be significantly different if their average ranks differ by at least one critical difference $CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$. Here, $q_\alpha = 3.031$ at significance level $\alpha = 0.05$, therefore $CD = 1.2924(k = 8, N = 66)$. The detailed CD diagrams are shown in Fig. 1. Holm test is based on the ranking of multiple classifiers. Firstly, the statistic $Z = \frac{Rank_i - Rank_j}{SE}$ obeying the standard normal distribution is calculated, where $SE = \frac{k(k+1)}{6N}$, $Rank_i$ is the average ranking of the i -th classifier on all data sets. Then the corresponding

probability p can be obtained according to Z and sorted in ascending order. Finally, the probability p and $\alpha/(k - i)$ are compared to judge whether the original assumption (the two classifiers have the same performance) is rejected. Table 9 summarizes the Holm test result.

Based on the results of comparison experiments, the following observations can be gained:

- LSLC can outperform other state-of-the-art for multi-label learning algorithms in some case of different label missing rates. According to the Tables 2, 3, 4, 5, 6, 7, and 8, we mainly analyze the experimental results from two directions. From the horizontal direction, we can see clearly that among all the evaluation matrices, the performance of LSLC and LSLCnew perform slightly better on most data sets than other comparison algorithms, and an analysis of Fig. 1 reveals that LPLC and LPLCnew are relatively left-sided in the evaluation matrices compared with other algorithms. From the vertical direction, the experimental results of MLKNN, LPLC, LIFT, LLSF and Glocal have obvious fluctuation with the increase of label missing rate. But the experimental results of algorithm LSML and LSLC show change relatively smooth and only rises slowly in a small range. The main cause of phenomenon is attributed to the algorithm itself. LPLC only considers positive label correlations. Both LIFT and LLSF consider only label-specific features. For MLKNN, LPLC and LLSF, they cannot handle missing labels. Although Glocal takes into

Table 9 The Holm test result

Evaluation metrics	i	Classifier	Target classifier	$(Rank_i - Rank_{LSLC})/SE$	Z	p	$\alpha/(k-i)$
HL	1	MLKNN	LSLC	(3.6970–5.5758)/(0.4264)	-4.4061	0.0000	0.00710
	2	LPLC	LSLC	(2.9848–5.5758)/(0.4264)	-6.0762	0.0000	0.00830
	3	LLSF	LSLC	(2.8636–5.5758)/(0.4264)	-6.3605	0.0000	0.01000
	4	LIFT	LSLC	(4.1667–5.5758)/(0.4264)	-3.3046	0.0010	0.01250
	5	Glocal	LSLC	(5.6515–5.5758)/(0.4264)	0.1777	0.8590	0.01670
	6	LSML	LSLC	(5.5152–5.5758)/(0.4264)	-0.1421	0.8870	0.02500
	7	LSLCnew	LSLC	(5.5455–5.5758)/(0.4264)	-0.0711	0.9433	0.05000
SA	1	MLKNN	LSLC	(6.0455–2.6364)/(0.4264)	7.9950	0.0000	0.00710
	2	LIFT	LSLC	(6.0303–2.6364)/(0.4264)	7.9595	0.0000	0.00830
	3	LLSF	LSLC	(4.5909–2.6364)/(0.4264)	4.5838	0.0000	0.01000
	4	Glocal	LSLC	(7.6742–2.6364)/(0.4264)	11.8149	0.0000	0.01250
	5	LPLC	LSLC	(3.8788–2.6364)/(0.4264)	2.9137	0.0036	0.01670
	6	LSML	LSLC	(2.6439–2.6364)/(0.4264)	0.0178	0.9858	0.02500
	7	LSLCnew	LSLC	(2.5–2.6364)/(0.4264)	-0.3198	0.7491	0.05000
MF1	1	MLKNN	LSLC	(6.3485–1.9848)/(0.4264)	10.2336	0.0000	0.00710
	2	LPLC	LSLC	(4.7121–1.9848)/(0.4264)	6.3960	0.0000	0.00830
	3	LIFT	LSLC	(5.803–1.9848)/(0.4264)	8.9544	0.0000	0.01000
	4	LLSF	LSLC	(4.9848–1.9848)/(0.4264)	7.0356	0.0000	0.01250
	5	Glocal	LSLC	(8–1.9848)/(0.4264)	14.1068	0.0000	0.01670
	6	LSLCnew	LSLC	(2.1212–1.9848)/(0.4264)	0.3198	0.7491	0.02500
	7	LSML	LSLC	(2.0455–1.9848)/(0.4264)	0.1421	0.8870	0.05000
AP	1	MLKNN	LSLC	(4.8636–2.3939)/(0.4264)	5.7920	0.0000	0.00710
	2	LPLC	LSLC	(6.5758–2.3939)/(0.4264)	9.8072	0.0000	0.00830
	3	LIFT	LSLC	(7.5152–2.3939)/(0.4264)	12.0103	0.0000	0.01000
	4	LLSF	LSLC	(5.4242–2.3939)/(0.4264)	7.1067	0.0000	0.01250
	5	Glocal	LSLC	(4.5909–2.3939)/(0.4264)	5.1524	0.0000	0.01670
	6	LSML	LSLC	(2.2576–2.3939)/(0.4264)	-0.3198	0.7491	0.02500
	7	LSLCnew	LSLC	(2.3788–2.3939)/(0.4264)	-0.0355	0.9717	0.05000
OE	1	MLKNN	LSLC	(6.1212–2.803)/(0.4264)	7.7818	0.0000	0.00710
	2	LIFT	LSLC	(7.2424–2.803)/(0.4264)	10.4113	0.0000	0.00830
	3	LLSF	LSLC	(5.7121–2.803)/(0.4264)	6.8224	0.0000	0.01000
	4	Glocal	LSLC	(4.697–2.803)/(0.4264)	4.4417	0.0000	0.01250
	5	LPLC	LSLC	(4.3485–2.803)/(0.4264)	3.6244	0.0003	0.01670
	6	LSLCnew	LSLC	(2.4697–2.803)/(0.4264)	-0.7817	0.4344	0.02500
	7	LSML	LSLC	(2.6061–2.803)/(0.4264)	-0.4619	0.6441	0.05000
RL	1	LPLC	LSLC	(6.9091–3.3939)/(0.4264)	8.2438	0.0000	0.00710
	2	LIFT	LSLC	(7.8939–3.3939)/(0.4264)	10.5534	0.0000	0.00830
	3	MLKNN	LSLC	(4.6061–3.3939)/(0.4264)	2.8427	0.0045	0.01000
	4	LLSF	LSLC	(3.1818–3.3939)/(0.4264)	-0.4975	0.6189	0.01250
	5	LSML	LSLC	(3.2424–3.3939)/(0.4264)	-0.3553	0.7223	0.01670
	6	LSLCnew	LSLC	(3.2424–3.3939)/(0.4264)	-0.3553	0.7223	0.02500
	7	Glocal	LSLC	(3.4394–3.3939)/(0.4264)	0.1066	0.9151	0.05000
AUC	1	LPLC	LSLC	(6.9242–3.5152)/(0.4264)	7.9950	0.0000	0.00710
	2	LIFT	LSLC	(7.8788–3.5152)/(0.4264)	10.2336	0.0000	0.00830
	3	MLKNN	LSLC	(4.6515–3.5152)/(0.4264)	2.6650	0.0077	0.01000
	4	LLSF	LSLC	(3.1515–3.5152)/(0.4264)	-0.8528	0.3938	0.01250
	5	Glocal	LSLC	(3.2424–3.5152)/(0.4264)	-0.6396	0.5224	0.01670
	6	LSML	LSLC	(3.3182–3.5152)/(0.4264)	-0.4619	0.6441	0.02500
	7	LSLCnew	LSLC	(3.3182–3.5152)/(0.4264)	-0.4619	0.6441	0.05000

account the fact that label correlations and missing labels, it cannot select label-specific features.

- It can be seen from Table 9 that the performance of LSLC is different from that of MLKNN, LPLC, LIFT, LLSF and Glocal classifiers. Although the

performance of LSLC and LSML is the same in holm test, it can be seen from the results in Fig. 1 that the performance of LSLC is better than that of LSML. LSML can handle missing labels, label correlations and label-specific features, but it only uses positive

Table 10 The Friedman statistic F_F and critical value

Evaluation metrics	FF critical value
HL	113.6970
SA	316.0947
MF1	392.1970
AP	311.9394 1.2924
OE	244.9596
RL	267.9848
AUC	269.2727

label correlations. In LSLC, first of all, the positive and negative label correlations can be learned from the incomplete training data sets, and then missing label will be recovered by learned label correlations, finally multi-label classification task will be modeled by selecting label-specific features iteratively.

- LSLCnew still shows better performance compared with LSLC and it is faster than LSLC according to experimental results of all data sets.

In conclusion, our proposed algorithm shows a competitive performance compared to other exist state-of-the-art algorithm for multi-label classification task with missing label (Table 10).

5.5 Convergence

In our proposed algorithm, the accelerated proximal gradient(APG) is used to quickly solve the objective loss

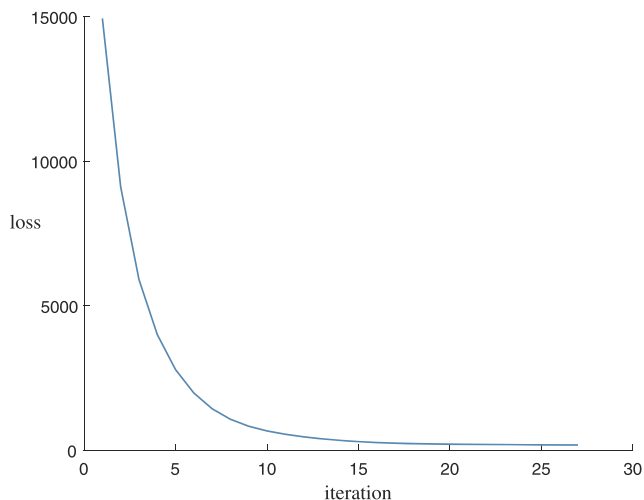


Fig. 2 Visualization of the loss function value w.r.t the number of iterations over ‘medical’ dataset

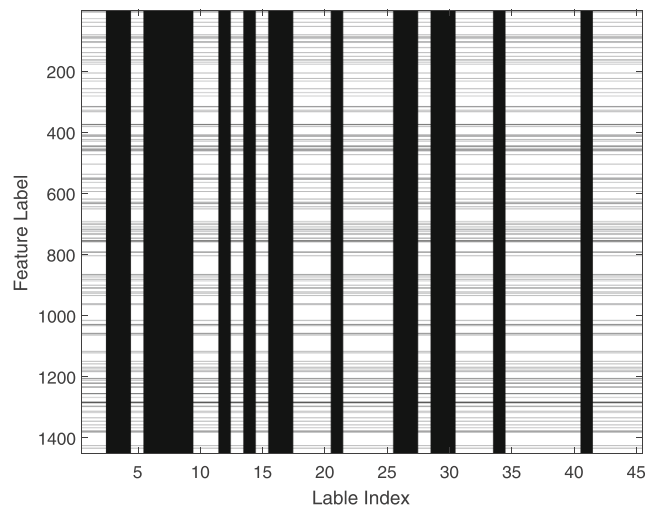


Fig. 3 Visualization of the label-specific features over ‘medical’ dataset

function. The loss function value w.r.t the number of iterations is shown in Fig. 2, and we conduct it over ‘medical’ data set. An analysis of Fig. 2 reveals that the ordinate value drops rapidly in the early stage over these two data sets and tends to be stable. Similarly, this phenomenon also occurs in other data sets.

5.6 Analysis of model parameters

In this section, we mainly analyze model parameters, that is, label-specific features and label correlations. Firstly, we conduct LSLC over ‘medical’ dataset with 60% label missing rate and visualize matrix \mathbf{W} , which is shown in

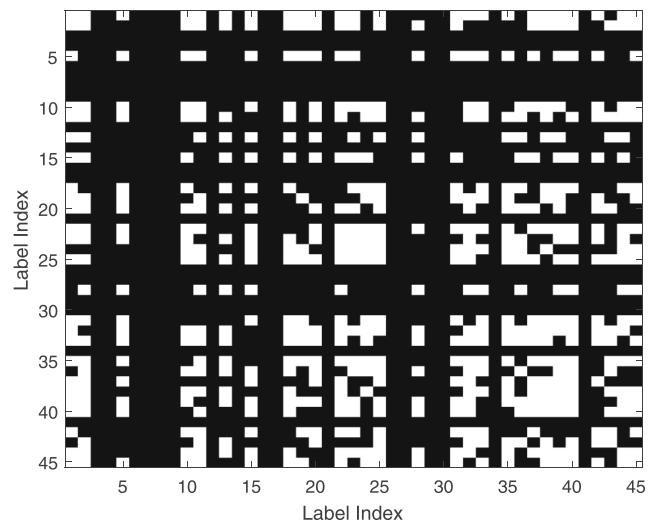


Fig. 4 Visualization of positive label correlations over ‘medical’ dataset

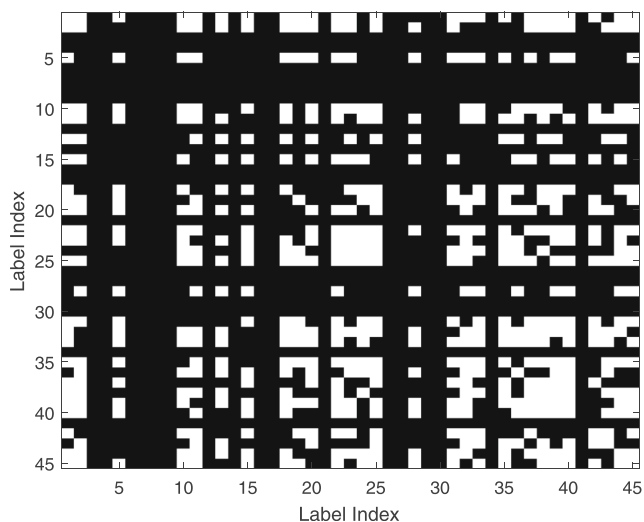


Fig. 5 Visualization of negative label correlations over ‘medical’ dataset

Fig. 3. Here, the horizontal axis and the vertical axis represent feature index and label index respectively. As can be seen from Fig. 3, we can select label-specific features represented by white color, and each class label is only determined by a certain subset of specific features from

original feature space. At the same time, label-specific feature selection is based on the assumption that the label matrix is complete. Thus, the label correlations play an important role in dealing with missing label task and it can be obtained by the learned model parameter \mathbf{P} and \mathbf{N} according to 3.4 and 3.5 respectively. Similarly, we conduct LSLC over ‘medical’ dataset and visualize \mathbf{P} and \mathbf{N} two matrices are shown in Figs. 4 and 5. Here, the horizontal axis and the vertical axis represent label index. Note, the points with white color indicate the nonzero entries of \mathbf{P} and \mathbf{N} . After observing the Fig. 4, we can arrive at the following conclusion that each label is only positive label correlation with some other labels. Likewise, this phenomenon also occurs in the Fig. 5.

To intuitively see the impact of positive and negative label correlations on classification results. We degenerate the algorithm of LSLC into two algorithms of LSLC-P and LSLC-N respectively. The former only positive label correlations are considered, while the latter only negative label correlations are considered. Simplicity, we conduct these algorithms over ‘medical’, ‘stack-chemistry’ and ‘rcv1subsets1’ datasets with 60% missing rate of class label. As shown in Table 11, it is effective to strengthen the

Table 11 The Experimental results of these models. For HL,RL and OE, the smaller the better; whereas for AP and AUC, the bigger the better. Best results are highlighted in bold

Dataset	Evaluation metrics	LSLC	LSLC-P	LSLC-N
rcv1subset1	HL	0.0285 ± 0.0001	0.0311 ± 0.0003	0.0312 ± 0.0003
	SA	0.0337 ± 0.0014	0.0253 ± 0.0025	0.0243 ± 0.0011
	MF1	0.3818 ± 0.0029	0.3125 ± 0.0048	0.3121 ± 0.0036
	AP	0.5360 ± 0.0025	0.4796 ± 0.0019	0.4785 ± 0.0019
	OE	0.4883 ± 0.0038	0.5301 ± 0.0033	0.5291 ± 0.0024
	RL	0.0971 ± 0.0007	0.145 ± 0.0014	0.1458 ± 0.0014
	AUC	0.8811 ± 0.0009	0.8303 ± 0.0013	0.8287 ± 0.0019
stackex-chemistry	HL	0.0144 ± 0.0001	0.0168 ± 0.0003	0.0170 ± 0.0003
	SA	0.0578 ± 0.0010	0.0367 ± 0.0003	0.0372 ± 0.0003
	MF1	0.3172 ± 0.0036	0.2482 ± 0.0013	0.2484 ± 0.0013
	AP	0.4113 ± 0.0024	0.3627 ± 0.0029	0.3628 ± 0.0029
	OE	0.5845 ± 0.0036	0.6258 ± 0.0051	0.6260 ± 0.0051
	RL	0.1622 ± 0.0006	0.2165 ± 0.0009	0.2143 ± 0.0009
	AUC	0.8273 ± 0.0007	0.7740 ± 0.0010	0.7753 ± 0.0010
medical	HL	0.0156 ± 0.0004	0.0203 ± 0.0007	0.0204 ± 0.0007
	SA	0.4976 ± 0.0079	0.4133 ± 0.009	0.4093 ± 0.009
	MF1	0.6569 ± 0.0072	0.6069 ± 0.0077	0.6101 ± 0.0077
	AP	0.8478 ± 0.006	0.7638 ± 0.0088	0.7692 ± 0.0088
	OE	0.1979 ± 0.0096	0.2872 ± 0.0116	0.2816 ± 0.0116
	RL	0.0366 ± 0.0027	0.0865 ± 0.0051	0.0816 ± 0.0051
	AUC	0.9555 ± 0.0035	0.9002 ± 0.0049	0.9034 ± 0.0049

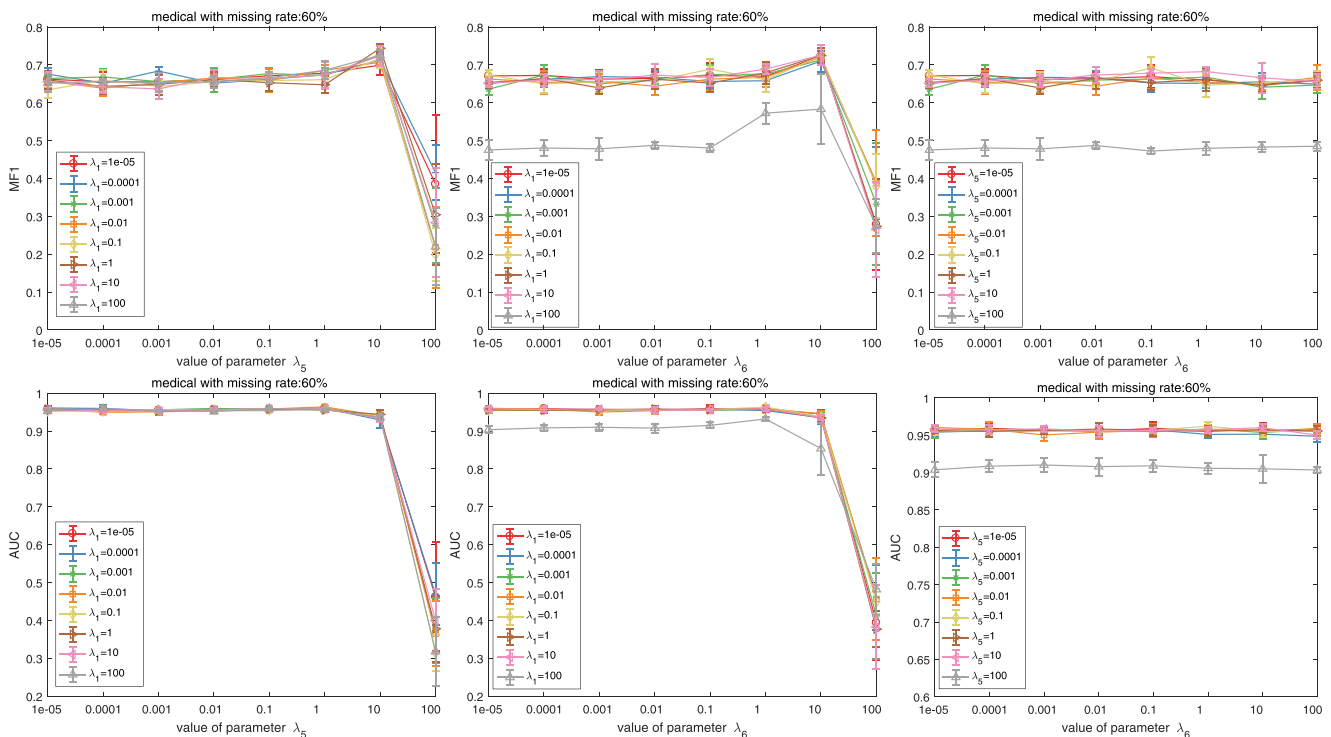


Fig. 6 The Sensitivity to parameters with 60% missing rate of class label

classification ability if the positive and negative label correlations are considered.

5.7 Sensitivity to parameters

In this section, we mainly analyze the three parameters λ_1 , λ_5 , and λ_6 , where λ_1 controls the sparsity of label-specific, λ_5 controls the positive label relationship and λ_6 controls the negative label relationship. We conduct the experiment on "medical" data set with 60% missing label, and dynamically changing two of parameters by fixing other four parameters with the optimal value respectively. We randomly selected 80% of the data as training data and the rest as test data. The experiment is repeated ten times and a five-fold cross test is used. The experimental results for evaluation metrics of MF1 and AUC are summarized in Fig. 6. An analysis of these Figures reveals that LSLC is sensitive obviously to these tradeoff parameters λ_1 , λ_5 , and λ_6 in the set rang. When any one of the three parameters λ_1 , λ_5 , and λ_6 exceeds a certain range, the performance of algorithm LSLC will be weakened.

6 Conclusion

In this paper, we have proposed a novel algorithm named LSLC, which can deal with labelspecific for multi-label learning with missing labels by taking both positive label correlations and negative label correlations into consideration. For

missing labels, LSLC can complete the missing label matrix through the positive and negative label correlations and then completion label matrix drives label-specific features selection. Besides, the APG is applied in our proposed algorithm to quickly solve the objective function iteratively. Furthermore, the experimental results show that both positive and negative correlation of labels plays a certain role in the final classification effect of the classifier. After a series of experiments in eleven benchmark data sets, our algorithm shows better performance and can be better select special label features from multi-label classification tasks with missing labels.

References

1. Zhou H, Zhao H, Zhang Y (2020) Nonlinear system modeling using self-organizing fuzzy neural networks for industrial applications[J]. *Appl Intell*:1–16
2. Bencherif A, Chouireb F (2019) A recurrent TSK interval type-2 fuzzy neural networks control with online structure and parameter learning for mobile robot trajectory tracking[J]. *Appl Intell* 49(11): 3881–3893
3. de Campos SPV, Guimaraes AJ, Araujo VS et al (2019) Incremental regularized Data Density-Based Clustering neural networks to aid in the construction of effort forecasting systems in software development[J]. *Appl Intell* 49(9):3221–3234
4. Bae JS, Oh SK, Pedrycz W et al (2019) Design of fuzzy radial basis function neural network classifier based on information data pre-processing for recycling black plastic wastes: comparative studies

- of ATR FT-IR and Raman spectroscopy[J]. *Appl Intell* 49(3):929–949
5. Ciarelli PM, Oliveira E, Salles EOT (2014) Multi-label incremental learning applied to web page categorization. *Neural Comput Appl* 24(6):1403–1419
 6. Boutell MR, Luo J, Shen X et al (2014) Learning multi-label scene classification. *Pattern Recognit* 37(9):1757–1771
 7. Alves RT, Delgado MR, Freitas AA (2008) Multi-label hierarchical classification of protein functions with artificial immune systems. In: *Brazilian Symposium on Bioinformatics*. Springer, Berlin, Heidelberg, pp 1–12
 8. Zhang ML, Zhou ZH (2007) ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognit* 40(7):2038–2048
 9. Read J, Pfahringer B, Holmes G et al (2011) Classifier chains for multi-label classification. *Mach Learn* 85(3):333–359
 10. Huang S-J, Zhou Z-H (2012) Multi-label learning by exploiting label correlations locally. In: *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pp 949–955
 11. Charte F, Rivera AJ, Del Jesus MJ et al (2014) LI-MLC: a label inference methodology for addressing high dimensionality in the label space for multilabel classification. *IEEE Trans Neural Netw Learn Syst* 25(10):1842–1854
 12. Nan G, Li Q, Dou R et al (2018) Local positive and negative correlation-based k-labelsets for multi-label classification. *Neurocomputing* 318:90–101
 13. Zhu Y, Kwok JT, Zhou ZH (2017) Multi-label learning with global and local label correlation. *IEEE Trans Knowl Data Eng* 30(6):1081–1094
 14. Tsoumakas G, Katakis I, Vlahavas I (2010) Random k-labelsets for multilabel classification. *IEEE Trans Knowl Data Eng* 23(7):1079–1089
 15. Huang J, Li G, Wang S et al (2017) Multi-label classification by exploiting local positive and negative pairwise label correlation. *Neurocomputing* 257:164–174
 16. Wu G, Tian Y, Liu D (2018) Cost-sensitive multi-label learning with positive and negative label pairwise correlations. *Neural Netw* 108:411–423
 17. Zhang C, Bi J, Xu S, Ramentol E, Fan G, Qiao B, Fujita H (2019) Multi-Imbalance: an open-source software for multi-class imbalance learning[J]. *Knowl-Based Syst* 174
 18. Sun J, Li H, Fujita H, Binbin F, Ai W (2020) Class-imbalanced dynamic financial distress prediction based on Adaboost-SVM ensemble combined with SMOTE and time weighting[J]. *Inform Fusion* 54
 19. Sun J, Lang J, Fujita H, Li H (2018) Imbalanced enterprise credit evaluation with DTE-SBD: decision tree ensemble based on SMOTE and bagging with differentiated sampling rates[J]. *Inform Sci* 425
 20. Doquire G, Verleysen M (2013) Mutual information-based feature selection for multilabel classification. *Neurocomputing* 122:148–155
 21. Lee J, Kim DW (2015) Mutual information-based multi-label feature selection using interaction information. *Expert Syst Appl* 42(4):2013–2025
 22. Wang C, Lin Y, Liu J (2019) Feature selection for multi-label learning with missing labels. *Appl Intell*:1–16
 23. Jorge G-L, Ventura S, Cano A (2019) Distributed selection of continuous features in multilabel classification using mutual information[J]. *IEEE Trans Neural Netw Learn Syst*
 24. Jorge G-L, Ventura S, Cano A (2020) Distributed multi-label feature selection using individual mutual information measures[J]. *Knowl-Based Syst* 188:105052
 25. Zhang ML, Wu L (2014) Lift: Multi-label learning with label-specific features. *IEEE Trans Pattern Anal Mach Intell* 37(1):107–120
 26. Huang J, Li G, Huang Q et al (2016) Learning label-specific features and class-dependent labels for multi-label classification. *IEEE Trans Knowl Data Eng* 28(12):3309–3323
 27. Xu S, Yang X, Yu H et al (2016) Multi-label learning with label-specific feature reduction. *Knowl-Based Syst* 104:52–61
 28. Wu B, Lyu S, Hu BG et al (2015) Multi-label learning with missing labels for image annotation and facial action unit recognition. *Pattern Recognit* 48(7):2279–2289
 29. Xu M, Jin R, Zhou ZH (2013) Speedup matrix completion with side information: Application to multi-label learning. In: *Advances in neural information processing systems*, pp 2301–2309
 30. Liu Y, Wen K, Gao Q et al (2018) SVM based multi-label learning with missing labels for image annotation. *Pattern Recogn* 78:307–317
 31. Zhu P, Xu Q, Hu Q, Zhang C, Zhao H (2018) Multi-label feature selection with missing labels. *Pattern Recogn* 74:488–502
 32. Tan Q, Yu G, Domeniconi C, Wang J, Zhang Z (2018) Incomplete multi-view weak-label learning. In *IJCAI*
 33. Dong HC, Li YF, Zhou ZH (2018) Learning from semi-supervised weak-label data. In: *AAAI*
 34. Guo Y, Chung F, Li G, et al (2019) Leveraging label-specific discriminant mapping features for multi-label learning[J]. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13(2):1–23
 35. Han H, Huang M, Zhang Y et al (2019) Multi-label learning with label specific features using correlation information. *IEEE Access* 7:11474–11484
 36. Weng W, Lin Y, Wu S et al (2018) Multi-label learning based on label-specific features and local pairwise label correlation. *Neurocomputing* 273:385–394
 37. Weng W, Chen Y-N, Chen C-L, Wu S-X, Liu J-H (2020) Non-sparse label specific features selection for multi-label classification[J]. *Neurocomputing* 377
 38. Liu J, Li Y, Weng W et al (2020) Feature selection for multi-label learning with streaming label[J]. *Neurocomputing*
 39. Huang J, Qin F, Zheng X et al (2019) Improving multi-label classification with missing labels by learning label-specific features. *Inf Sci* 492:124–146
 40. Beck A, Teboulle M (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences* 2(1):183–202
 41. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets[J]. *J Mach Learn Res* 7(Jan):1–30
 42. Hühn JC, Hüllermeier E (2008) FR3: A fuzzy rule learner for inducing reliable classifiers[J]. *IEEE Transactions on Fuzzy Systems* 17(1):138–149

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.