Soft Computing
https://doi.org/10.1007/s00500-020-04775-1

FOUNDATIONS

Joint label completion and label-specific features for multi-label learning algorithm

Yibin Wang[1,2] · Weijie Zheng[1] · Yusheng Cheng[1,2] · Dawei Zhao[1]

Abstract

Label correlations have always been one of the hotspots of multi-label learning. Using label correlations to complete the original label can enrich the information of the label matrix. At the same time, label-specific features give a thought that different labels have inherent characteristics that can be distinguished, and we can use label correlations to enhance the learning process of label-specific features among similar labels. At present, most of the algorithms combine label correlations and label-specific features to improve the multi-label learning effect, but do not consider the impact of label marking errors or defaults in data sets. In fact, the label completion method can further enrich the information of label matrix, and then the joint learning framework of joint label-specific features can effectively improve the robustness of the multi-label learning algorithm. Based on this, this paper proposes a multi-label learning algorithm for joint label completion and label-specific features, and constructs a new multi-label learning algorithm framework by means of joint label completion and label-specific features. Completion matrix and label-specific features are obtained by alternating iteration method, and the label matrix updating the optimization framework fully considers the label correlations. The algorithm in this paper has been demonstrated and trained on several benchmark multi-label data sets by extensive experiments, which verifies the effectiveness of the algorithm.

Keywords Multi-label learning · Label correlations · Label completion · Label-specific features

1 Introduction

Multi-label learning is one of the main frameworks for dealing with real-world objects with rich semantics. Numerous studies have shown that considering the label correlations and label-specific features can effectively improve the performance of multi-label learning. At present, most multi-label learning methods pay enough attention to the label correlations, but ignore the importance of label-specific features. For example, Zhang et al. first proposed LIFT (Zhang and Wu 2015), a research algorithm on label-specific features, which means that, firstly, k-means clustering is used to map features, then label samples are classified into positive and negative classes according to the relevant features of labels. Finally, SVM (Zendehboudi et al. 2018) is used to classify multi-labels, but this algorithm ignores label correlations. Xu et al. (2016) proposed a multi-label learning algorithm based on fuzzy rough sets. The importance of features was studied by using fuzzy rough sets, and the reduction of label-specific features was realized by former greedy search method. Huang et al. (2018) proposed JFSC algorithm, which minimized the distance of each label through a Fisher discriminant-based regular term method, and obtained the second-order relationship of labels by constraining the similarity of any two single-label model parameters. Huang et al. (2016) proposed a multi-label learning algorithm (LLSF-DL) for label-specific features and class-independent labels, by adding l_1 regular terms as sparse constraints, and using cosine similarity to consider the label correlations. However, when classifying numerical features, many experiments are needed to adjust the parameters.

Communicated by A. Di Nola.

✉ Yusheng Cheng
chengyshaq@163.com

[1] School of Computer and Information, Anqing Normal University, Anqing 246011, Anhui, China

[2] The University Key Laboratory of Intelligent Perception and Computing of Anhui Province, Anqing 246133, China

Published online: 24 February 2020

Label-specific features can extract pivotal characteristics corresponding to each label. If the correlation coefficient between labels is large, the pivotal characteristics extracted by label-specific features have a high similarity. Therefore, the introduction of label correlations learning into the algorithm with learning label-specific features can effectively improve the learning efficiency of label-specific features.

In dealing with the label correlations, the method of cosine similarity between labels, covariance matrix, and information entropy is mainly used to add label correlations matrix directly into the algorithm as a priori knowledge. For example, Huang et al. (2015) proposed to learn multi-label classification of label-specific features by calculating cosine similarity of labels. If two labels have strong correlation, they can share label-specific features, which have a good effect in multi-label classification. However, the real label correlations matrix is not necessarily symmetrical. Han et al. (2019) consider learning feature space and label space correlation information for each label, but the label correlations matrix still uses cosine similarity. Nguyen et al. (2019) proposed that the correlation of labels calculated by Bayesian formula is a priori condition of the algorithm, which improves the performance of the algorithm. Although label correlations as a priori knowledge of the algorithm can reduce the time complexity of the algorithm, it is generally difficult to determine the label correlations matrix in advance, so the performance improvement of the algorithm as a priori knowledge may be insufficient.

The above methods are usually for complete multi-label data sets. However, most of the existing multi-label data sets are labeled manually, so labeling will inevitably lead to errors or loss of label information. It is undeniable that mislabeling or default labeling may lead to the loss of important information, so it is particularly important to complete the labels of incomplete multi-label data sets.

At present, the most commonly used method is to use the correlation to complete the original labels. For example, Cheng et al. (2018) proposed that hidden information in feature space could be obtained by means of mean shift clustering method, and a non-equilibrium label completion method was used to obtain label completion matrix by measuring the correlation of labels with information entropy. Xu et al. (2014) proposed to learn the low-rank correlation matrix of labels through the algorithm framework and completed the label matrix with this matrix. However, when learning the low-rank correlation matrix of labels, the attributes of the learning class were not taken into account. Guo et al. (2019) proposed a label completion method based on deep convolution neural network (Zhao et al. 2019). By learning a small number of labeled data sets, the labeled data can be completed, and the noise labeled data can be corrected to improve the performance of the algorithm. Liu et al. (2018) put forward the assumption of smoothness of label data based on matrix completion model, that is, adjacent instances should have a set of similar labels, making full use of the label information. Zhang et al. (2018) proposed joint learning and assign weights separately to the relevance of the label-specific features and label correlations. However, the incompleteness of the original label space is not considered in the algorithm framework, which affects the classification performance. He et al. (2019) proposed to learn high-order asymmetric label correlations matrix to deal with multi-label classification of defective labels, but did not consider dimensionality reduction of label space. Huang et al. (2019) learned high-order label correlations by algorithm, and obtained a new label complementary matrix, which has a good effect in the classification of defective multi-label data sets. Chen et al. (2013) proposed fast label algorithm to complete the incomplete label matrix and enrich the information by learning regularization parameters. Sun et al. (2010) proposed WELL algorithm to obtain similarity matrix in feature space, and learned each label of it to complete the label matrix. Wu et al. (2015) proposed ML–MG algorithm, which takes the semantic levels of label correlations and label-specific features as undirected and directed edges, respectively, through mixed graphs, and completes the label matrix with graph theory knowledge. Dong et al. (2018) proposed a semi-supervised multi-label learning algorithm for SSWL, which completes the label matrix by considering the correlation between data and the label correlations.

Although label correlations are used to complete labels, which makes multi-label learning get better classification effect, most of the research on label completion is based on the label correlations as a priori condition to complete the original label matrix. It can be seen that the joint learning of label completion and label-specific features can take label correlations into account while learning them. The combination of the two can effectively improve the performance of classifier.

Based on this, this paper firstly establishes an optimization framework to model the learning of label-specific features, and uses the label correlations matrix learned by the algorithm to complete the label matrix. In order to distinguish the original label matrix from the complementary one, a difference matrix is introduced as the restriction condition of the algorithm framework, and a joint learning framework is constructed by sparse feature space and difference matrix of the l_1 norm and the $l_{2,1}$ norm, respectively. In this paper, a joint label completion and label-specific features for multi-label learning algorithm (JLCLS) is proposed to learn label-specific features while completing the original label space by using the correlation matrix between learning labels, and to improve the learning performance of label-specific features by using the correlation matrix of labels as auxiliary information. The JLCLS algorithm framework takes full account of the label correlations to enrich the label-specific features information, and then puts the completed label matrix into the algorithm framework to learn the label-specific features and label completion matrix jointly, so as to solve the multi-label classification learning problem.

The rest of this article is organized as follows. Section 2 gives a description of the multi-label theory and problems, the modeling process of the joint learning framework and the methods and iterative formulas for solving the parameters in the model. Section 3 introduces the joint learning framework algorithm flow and algorithm complexity analysis proposed in this paper. In Sect. 4, JLCLS is compared with other multi-label learning algorithms on the multi-label and missing label data sets, and the experimental results are analyzed, so as to prove the necessity of putting forward the framework of joint label completion and label-specific learning. This paper analyzes the influence of parameters on the experimental results in the joint learning framework through parameter sensitivity experiments, and analyzes the difference between the algorithm and other multi-label algorithms through statistical hypothesis testing in Sect. 5. In the last section, this paper sums up what has been discussed and put forward further research.

2 Joint learning framework modeling and optimization

2.1 Multi-label theory and problem description

In multi-label learning, there are input training data $\boldsymbol{X}$ and label matrix $\boldsymbol{Y}$, where $\boldsymbol{X} \in R^{n \times d}$, n is the number of data sets, d is the features number of data sets, $\boldsymbol{Y} \in R^{n \times l}$, l is the number of labels. $S = \{(x_i, y_i) | 1 \leq i \leq n\}$ is a multi-label training set, including $\boldsymbol{x_j} = \{x_{i1}, x_{i2}, \ldots x_{id}\}$ is a features vector, $\boldsymbol{y_i} = \{y_{i1}, y_{i2}, \ldots y_{il}\}$ is a label vector, tabbed learning task is to find a mapping: $f : X \rightarrow 2^Y$.

For the label matrix $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots \boldsymbol{y}_l\}$, where $\boldsymbol{y_j}$ is the label vector of jth label, most label completion studies now to multiply the label matrix $\boldsymbol{Y}$ by $\boldsymbol{A}$ label correlations matrix $\boldsymbol{A}$, which is to make $\boldsymbol{A}$ linear transformation (Fu et al. 2013) of the label vector in the label matrix to obtain a new label matrix.

Nowadays, most labels are based on manual ones, so it is difficult to avoid the omission and misclassification of labels. Therefore, the original label space can be filled by label correlations. For example, in the following two figures, Fig. 1a labels "sky" without "ocean" and Fig. 1b labels "green space" without "sky". Intuitively, "sky" and "ocean" have strong label correlations, so considering the correlation between them, label completion is carried out for (a) and (b) in Fig. 1, respectively, which is conducive to improving the performance of multi-label learning classification algorithm.

$$\begin{aligned}&\boldsymbol{Y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} 1 & 0.7 & 0.6 \\ 0.6 & 1 & 0.2 \\ 0.5 & 0.3 & 1 \end{bmatrix}, \\ &\boldsymbol{YA} = \begin{bmatrix} 1 & 0.7 & 0.6 \\ 0.5 & 0.3 & 1 \end{bmatrix} = \boldsymbol{Y}^*\end{aligned} \tag{1}$$

The columns of the matrix $\boldsymbol{Y}$ represent "sky", "ocean" and "green space" in turn, and the rows of the matrix $\boldsymbol{Y}$ represent (a) and (b) in Fig. 1 in turn. Matrix $\boldsymbol{A}$ represents the completion matrix with label correlations. The rows and columns of this matrix represent "sky," "ocean," and "green space" in turn. Since the original label matrix is completed by utilizing the label correlations, we consider improving the performance of multi-label algorithm by combining the label correlations with the label-specific features. Because the two labels of "ocean" and "sky" have a strong correlation, they can have similar label-specific features, whereas the labels have distinguishing features.

2.2 Modeling of the joint learning framework

In the traditional classification algorithm model, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are usually placed in the algorithm model, the weight $\boldsymbol{W}$ is introduced, and a regular term is added to prevent overfitting of the algorithm model. The algorithm model is:

$$L(W) = \min_{W} \|XW - Y\|_F^2 + \frac{\beta}{2}\|W\|_F^2 \tag{2}$$

$L(\cdot)$ is the loss function, β is the regularization parameter, and $\boldsymbol{W} \in R^{d \times l}$. Next, we continue to complete the label matrix $\boldsymbol{Y}$. In order to make the label matrix different from the matrix after linear transformation, a difference matrix $\boldsymbol{Q}$ is introduced. The algorithm model becomes:

$$\begin{aligned}&\min_{W,A,Q} \|XW - YA\|_{\mathrm{F}}^2 + \frac{\beta}{2}\|W\|_F^2 + \theta\|Q\|_{2,1} \\ &\text{s.t.} \quad Y = YA + Q\end{aligned} \tag{3}$$

$\boldsymbol{A} \in R^{l \times l}$, $\boldsymbol{Q} \in R^{n \times l}$, and θ is a parameter of the model. From the above algorithm model, we can see that the completion matrix is constantly changing, and the matrix has the label correlations, which means that the label correlations will be taken into account when learning label-specific features. For the difference matrix $\boldsymbol{Q}$, defined

(**a**) sky. (**b**) grass.

Fig. 1 **a** “sky” and **b** “grass”

$\|Q\|_{2,1}=\sum_{j=1}^{l}\sqrt{\sum_{i=1}^{n}\left(Q_{ij}\right)^2}$, through matrix $l_{2,1}$ norm sparse label matrix after completion. $\boldsymbol{A}\in R^{l\times l}$ is a low-rank label correlation matrix, and a_{ij} represents the correlation between label y_i and y_j. The pairwise label correlation between each pair of class labels is contained by calculating the Euclidean distance between any pair of coefficient vectors of label correlation.

$$\mathcal{R}=\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}a_{ij}\left\|w_i-w_j\right\|_2^2 \tag{4}$$

From the above Eq. (4), the label-specific features are determined by the vector $\boldsymbol{w_i}$, and the value range of i is $1\le i\le l$. It can be found from Eq. (4) that if the labels $\boldsymbol{y_i}$ and $\boldsymbol{y_j}$ are strongly correlated, they will have similar label-specific features, and the corresponding model coefficients $\boldsymbol{w_i}$ and $\boldsymbol{w_j}$ will be quite similar. Otherwise, $\boldsymbol{w_i}$ and $\boldsymbol{w_j}$ will be dissimilar. The sparsity of label-specific features is represented by l_1 norms. In this way, the algorithm learns label-specific features. Considering all label vectors, the algorithm model is finally changed into:

$$\begin{aligned}&\min_{W,A,Q}\|XW-YA\|_F^2+\frac{\alpha}{2}\operatorname{tr}\left(AW^TW\right)+\frac{\beta}{2}\|W\|_1+\theta\|Q\|_{2,1}\\&\quad\text{s.t.}\quad Y=YA+Q\end{aligned} \tag{5}$$

$\boldsymbol{W}=(\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots\boldsymbol{w}_l)\in R^{d\times l},\boldsymbol{A}\in R^{l\times l}$, $\boldsymbol{Y}=(\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots\boldsymbol{y}_l)\in R^{n\times l},\alpha\ge 0,\beta\ge 0$ and $\theta\ge 0$ are parameters of the model. It can be seen from the above algorithm model that when the label matrix is completed, it will be affected by the weight $\boldsymbol{W}$. What influences the weight $\boldsymbol{W}$ is not only the classification model after the completion of the label matrix, but also the label correlations and label-specific features.

2.3 Optimization solution of joint learning model parameters

In the optimization process, this paper uses the idea of alternating iteration to solve matrix $\boldsymbol{A}$ and weight $\boldsymbol{W}$ corresponding to linear transformation. When given the matrix $\boldsymbol{A}$ and the difference matrix $\boldsymbol{Q}$, only the variable $\boldsymbol{W}$, then the algorithm model becomes:

$$\min_{W}\|XW-YA\|_F^2+\frac{\alpha}{2}\operatorname{tr}\left(AW^TW\right)+\frac{\beta}{2}\|W\|_1 \tag{6}$$

Since the regular term of l_1 norm is non-smooth, the objective function Eq. (6) is non-smooth. For this reason, this paper uses the proximal gradient descent method (Beck and Teboulle 2009a) to deal with the non-smooth objective function. So, the model becomes:

$$\min_{W\in H}F(W)=f(W)+g(W) \tag{7}$$

$$f(W)=\|XW-YA\|_F^2+\frac{\alpha}{2}\operatorname{tr}\left(AW^TW\right) \tag{8}$$

$$g(W)=\frac{\beta}{2}\|W\|_1 \tag{9}$$

H is the Hilbert space. $f(W)$ and $g(W)$ are convex functions. $f(W)$ satisfies Lipschitz condition, that is, $\|\nabla f(W_1)-\nabla f(W_2)\|\le L_f\|\Delta W\|$, and L_f is Lipschitz constant. In the process of proximal gradient descent, not directly to minimize $F(W)$, need to introduce $Q\left(W,W^{(t)}\right)$ quadratic approximation $F(W)$. So, the definition of $Q\left(W,W^{(t)}\right)$ is:

$$\begin{aligned}Q\left(W,W^{(t)}\right)=&f\left(W^{(t)}\right)+\left\langle\nabla f\left(W^{(t)}\right),W-W^{(t)}\right\rangle\\&+\frac{L_f}{2}\left\|W-W^{(t)}\right\|_F^2+g(W)\end{aligned} \tag{10}$$

Let

$$G^{(t)}=W^{(t)}-\frac{1}{L_f}\nabla f\left(W^{(t)}\right). \tag{11}$$

$$W = \arg\min_{W} Q\left(W, W^{(t)}\right) = \arg\min_{W} g(W) + \frac{L_f}{2}\left\|W - G^{(t)}\right\|_F^2 \\ = \arg\min_{W} \frac{1}{2}\left\|W - G^{(t)}\right\|_F^2 + \frac{\beta}{L_f}\|W\|_1 \tag{12}$$

In literature Beck and Teboulle (2009b), it is given that:

$$W^{(t)} = W_t + \frac{b_{t-1} - 1}{b_t}(W_t - W_{t-1}) \tag{13}$$

Sequence b_t satisfying $b_{t+1}^2 - b_{t+1} \leq b_t^2$ can improve the convergence rate to $O(t^{-2})$, and $\boldsymbol{W_t}$ is the result of $\boldsymbol{W}$ at the tth iteration.

Each iteration can be optimized by the following methods:

$$W_{t+1} = S_\varepsilon\left[G^{(t)}\right] = \arg\min_{W} \varepsilon\|W\|_1 + \frac{1}{2}\left\|W - G^{(t)}\right\|_F^2 \tag{14}$$

$S_\varepsilon[\cdot]$ is a soft-thresholding operator. For each element x_{ij} and $\varepsilon = \frac{\beta}{2L_f}$, this function can be defined as:

$$S_\varepsilon\left(x_{ij}\right) = \begin{cases} x_{ij} - \varepsilon & \text{if } x_{ij} > \varepsilon \\ x_{ij} + \varepsilon & \text{if } x_{ij} < -\varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

Lipschitz constant is calculated as follows. Given $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$, Lipschitz condition is satisfied according to $f(W)$:

$$\nabla f(W) = X^T XW - X^T Y + \frac{\alpha}{2}\alpha WA + \frac{\alpha}{2}WA^T \tag{16}$$

$$\|\nabla f(W_1) - \nabla f(W_2)\|_F^2 = \left\|X^T X\Delta W + \frac{\alpha}{2}\Delta WA + \frac{\alpha}{2}\Delta WA^T\right\|_F^2 \\ \leq 2\left\|X^T X\right\|_2^2\|\Delta W\|_F^2 + \|\alpha A\|_2^2\|\Delta W\|_F^2 + \left\|\alpha A^T\right\|_2^2\|\Delta W\|_F^2 \\ = \left(2\left\|X^T X\right\|_2^2 + \|\alpha A\|_2^2 + \left\|\alpha A^T\right\|_2^2\right)\|\Delta W\|_F^2 \tag{17}$$

In Eq. (17), $\Delta\boldsymbol{W} = \boldsymbol{W}_1 - \boldsymbol{W}_2$. So Lipschitz constant is:

$$L_f = \sqrt{2\|X^T X\|_2^2 + \|\alpha A\|_2^2 + \|\alpha A^T\|_2^2} \tag{18}$$

When given the weight $\boldsymbol{W}$, there are two variables $\boldsymbol{A}$ and $\boldsymbol{Q}$. So, the optimization model becomes:

$$\min_{A,Q}\|XW - YA\|_F^2 + \frac{\alpha}{2}\operatorname{tr}\left(AW^T W\right) + \theta\|Q\|_{2,1} \\ \text{s.t. } Y = YA + Q \tag{19}$$

In order to reflect the completion of the original label matrix and consider the label correlations, so as to interact, we need to introduce an auxiliary variable $\boldsymbol{U}$. Currently, the optimization model becomes:

$$\min_{A,U,Q}\|XW - YA\|_F^2 + \frac{\alpha}{2}\operatorname{tr}\left(UW^T W\right) + \theta\|Q\|_{2,1} \\ \text{s.t. } Y = YA + Q, \quad A = U \tag{20}$$

By using the inexact augmented Lagrange multiplication (IALM) (Rodriguez and Wohlberg 2016) to add qualifications to the optimization model, the optimization model becomes:

$$\min_{A,U,Q,\mu_1,\mu_2}\|XW - YA\|_F^2 + \frac{\alpha}{2}\operatorname{tr}\left(UW^T W\right) + \theta\|Q\|_{2,1} \\ + \frac{\rho}{2}\left\|Y - YA - Q + \frac{\mu_1}{\rho}\right\|_F^2 - \frac{1}{2\rho}\|\mu_1\|_F^2 + \frac{\rho}{2}\left\|A - U \right. \\ \left. + \frac{\mu_2}{\rho}\right\|_F^2 - \frac{1}{2\rho}\|\mu_2\|_F^2 \tag{21}$$

In Eq. (21), μ_1 and μ_2 are Lagrange multipliers, ρ is the parameter of the qualification. The following solution $\boldsymbol{U^{k+1}}$:

$$U^{k+1} = \frac{\mu_2^k}{\rho} + A^k + \frac{\alpha}{2\rho}\left(W^T W\right) \tag{22}$$

From the above formula, the linear transformation will be affected by the label correlations and label-specific features. The following solution $\boldsymbol{A^{k+1}}$:

$$A^{k+1} = \left[(2+\rho)Y^T Y - \rho I_3\right]^{-1} \\ \left[2Y^T XW + \rho\left(Y^T Y - Y^T Q^k + U^{k+1}\right) + \mu_1^k Y^T - \mu_2^k\right] \tag{23}$$

In Eq. (23), $\boldsymbol{I}_3$ is the identity matrix of $l \times l$. According to the minimization method of $2,1$ norm (Liu et al. 2010), to solve the $\boldsymbol{Q^{k+1}}$:

$$Q^{k+1}(:i) = \begin{cases} \dfrac{t_i^k - \dfrac{\theta}{\rho}}{t_i^k} & \text{if } t_i^k > \dfrac{\theta}{\rho} \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

Let $T^k = Y - YA^{k+1} + \frac{\mu_1^k}{\rho}$, t_i^k refers to the ith column of $\boldsymbol{T}$, $Q^{k+1}(:i)$ refers to the ith column of the difference matrix $\boldsymbol{Q^{k+1}}$.

The Lagrange multiplier $\boldsymbol{\mu_1^{k+1}}$, $\boldsymbol{\mu_2^{k+1}}$ is calculated as follows:

$$\mu_1^{k+1} = \mu_1^k + \rho\left(Y - YA^{k+1} - Q^{k+1}\right), \mu_2^{k+1} \\ = \mu_2^k + \rho\left(A^{k+1} - U^{k+1}\right) \tag{25}$$

The iterative update matrix $\boldsymbol{A}$, the difference matrix $\boldsymbol{Q}$, and the Lagrangian multiplier $\boldsymbol{\mu_1^{k+1}}, \boldsymbol{\mu_2^{k+1}}$ are closed. Iterative formula shows that each variable affects each other.

3 Joint learning model and complexity analysis

3.1 Multi-label learning algorithm for joint learning model

In this section, we propose the design idea of the algorithm. The original label matrix will be completed with the label correlations matrix, and the completed label matrix will be put into the optimization model. The high-order correlation matrix between labels was acquired by iterating each variable in the model, and the label completion matrix was put into the optimization model for joint learning with label-specific features. When optimizing the model, we need to introduce a difference matrix to make the completed matrix different from the original label one. This paper uses alternating iterative methods to solve various parameters.

```
Algorithm 1. The problem of Eq. (6) is solved by proximal gradient descent.
Input：Training data matrix $X$,label matrix $Y$， and weighting parameters $\alpha,\beta,\gamma$.
Output：$W$.
Initialization：$A = I, b_0 = b_1 = 1, W_0 = W_1 = (X^T X + \gamma I)^{-1} X^T Y A$
while not converged do
    $W^{(t)} = W_t + \frac{b_{t-1}-1}{b_t}(W_t - W_{t-1})$
    $G^{(t)} = W^t - \frac{1}{L_f}\nabla f\left(W^{(t)}\right)$
    $L_f = \sqrt{2||X^T X||_2^2 + ||\alpha A||_2^2 + ||\alpha A^T||_2^2}$
    $W_{t+1} = S_{\frac{\beta}{2L_f}}\left(G^{(t)}\right)$
    $b_{t+1} = \frac{1+\sqrt{4b_t^2+1}}{2}, t = t + 1$
end while
```

```
Algorithm 2. The problem of IALM in Eq. (21) is solved
Input：Training data matrix $X$,label matrix $Y$,weight matrix $W$,and weighting parameters $\alpha$，$\beta$，$\theta$
Output：$A, U, Q$
Initialization：$U = A = \mathbf{0}, Q = \mathbf{0}, \mu_1 = \mathbf{0}, \mu_2 = \mathbf{0}, \rho = 10^{-6}, max_\rho = 10^{10}, \lambda = 1.1, \epsilon = 10^{-6}$
while not converged do
    fix $A, Q$ and update variable $U$ , according to (22)
    fix $U, Q$ and update variable $A$ according to (23)
    fix $U, A$ and update variable $Q$ according to (24)
    fix $A, U, Q$ and update the multipliers $\mu_1$ and $\mu_2$ according to (25)
    update the parameter $\rho$ by $\rho = \min\left(\rho\lambda, max_\rho\right)$
    check the convergence conditions:
    $||Y - YA - Q||_\infty < \epsilon, ||U - A||_\infty < \epsilon$
end while
```

```
Algorithm 3 The JLCLS Framework
Input：Training data set $\{X,Y\}$，and weighting parameters $\alpha$，$\beta$，$\theta$
Output：$W,A$
Initialization：$A = I$
repeat
    fix $A$,$Q$ and update $W$ according to Algorithm 1
    fix $W$ and solve for $A$ using Algorithm 2
until convergence
```

3.2 JLCLS model algorithm complexity analysis

In the proposed algorithm, $\boldsymbol{X} \in R^{n\times d}, \boldsymbol{Y} \in \{0,1\}^{n\times l}, \boldsymbol{A} \in R^{l\times l}, \boldsymbol{W} \in R^{d\times l}$ where n represents the number of samples, d represents the number of features, and l represents the number of labels. In Algorithm 1, the most time consuming is Step 3 and Step 4. In step 3, $G^{(t)}$ is an intermediate variable, $f(\cdot)$ represents a gradient, and the algorithm complexity of calculating $\boldsymbol{W}$ is $O(nd^2l + nl^2)$. In Step 4, the algorithm complexity of calculating the Lipschitz constant is $O(d^3 + l^3)$. In Algorithm 2, the most time consuming is Step 3, Step 4 and Step 5. In Step 3, the algorithm complexity of calculating $\boldsymbol{U}$ is $O(nd^2l^2 + d^2l^2)$. In Step 4, the algorithm complexity of calculating $\boldsymbol{A}$ is $O(n^2l^2 + n^2d^2l)$. In Step 5, the algorithm complexity of calculating $\boldsymbol{Q}$ is $O(nl^2)$. In summary, the algorithm complexity of the JLCLS model is $O((n+1)(d^2l^2 + nl^2 + nd^2l) + d^3 + l^3)$. In this paper, the complexity of JLCLS joint learning algorithm is compared with LLSF and LLSF-DL. The calculation of Lipschitz constant has the same algorithm complexity $O(d^3 + l^3)$. According to the literature Huang et al. (2016), the algorithm complexity of LLSF and LLSF-DL is $O(d^2 + dl + l^2 + nd + nl)$. By comparison, the algorithm complexity of JLCLS, LLSF and LLSF-DL is very small, but in experiment, JLCLS is better than LLSF and LLSF-DL in most multi-label data sets.

4 Experiments

4.1 Data sets

In order to illustrate the effectiveness of the proposed algorithm, 12 data sets were selected, as shown in Table 1.

4.2 Comparison methods

In order to evaluate the performance of the proposed algorithm, several advanced multi-label classification algorithms were selected as comparison, among which are Rank-SVM (Elisseeff and Weston 2002), ML-KNN (Zhang and Zhou 2007), LLSF (Huang et al. 2015), LLSF-DL (Huang et al. 2016), LIFT (Zhang and Wu 2015). In addition, this paper learns from JLCLS algorithm that $\boldsymbol{w_i}$ belongs to label $\boldsymbol{y_i}$. These matrices $\boldsymbol{W}$ with label-specific features and the original training set of data features form a new label training set, which is put into the ELM (Huang et al. 2006) and BSVM (Sitompul and Nababan 2018) classification model, and JLCLS-ELM and JLCLS-SVM multi-label learning algorithms are obtained. In this paper, the parameter range of JLCLS algorithm is set to $\alpha \in [2^{-10}, 2^{10}], \beta \in [2^{-10}, 2^{10}], \gamma = \{0.1, 1, 10\}, \theta \in [2^{-10}, 2^{10}]$. The search range of the algorithm JLCLS-ELM parameters is $\alpha \in \left[2^{-6}, 2^{-1}\right], \beta \in \left[2^{-5}, 2^{-1}\right]$. The search range of the algorithm JLCLS-SVM parameters is $\alpha \in \left[2^{-6}, 2^{-1}\right], \beta \in \left[2^{-5}, 2^{-1}\right]$.

Table 1 Description of data sets

Data set	Instance	Label	Feature	LCard	Domain
Flags[b]	194	7	19	3.392	Image
Emotions[b]	593	6	72	1.869	Music
Birds[b]	645	20	260	1.471	Image
Genbase[b]	662	27	1185	1.2525	Biology
Medical[b]	978	45	1449	1.245	Text
Enron[a]	1702	53	1001	4.275	Text
Yeast[b]	2417	14	103	4.237	Biology
Arts[a]	5000	26	462	1.636	Text
Computers[a]	5000	33	681	1.508	Text
Education[a]	5000	33	550	1.461	Text
Science[a]	5000	40	743	1.451	Text
Society[a]	5000	27	636	1.692	Text

[a] Yahoo Web Pages (http://www.kecl.ntt.co.jp/as/members/ueda/yahoo.tar)

[b] Mulan (http://mulan.sourceforge.net/datasets-mlc.html)

Table 2 HL results of different algorithms on 12 datasets

Data set	JLCLS	JLCLS-ELM	JLCLS-SVM	RANK-SVM	ML-KNN	LLSF	LLSF-DL	LIFT
Genbase	**.001 ± .001**	**.001 ± .001**	**.001 ± .001**	.064 ± .006	.004 ± .003	**.001 ± .000**	**.001 ± .001**	.002 ± .001
Emotions	.193 ± .022	**.187 ± .021**	.192 ± .015	.390 ± .019	.194 ± .014	.202 ± .019	.188 ± .019	.197 ± .019
Medical	.010 ± .002	**.009 ± .002**	.011 ± .002	.038 ± .003	.015 ± .002	.012 ± .001	.010 ± .002	.013 ± .002
Arts	.053 ± .001	**.052 ± .002**	.053 ± .001	.075 ± .019	.057 ± .002	.054 ± .002	.054 ± .003	.053 ± .003
Computers	.034 ± .002	**.032 ± .002**	.035 ± .020	.044 ± .002	.035 ± .002	.034 ± .002	.037 ± .002	.033 ± .002
Education	.037 ± .001	**.036 ± .001**	.038 ± .001	.056 ± .002	.038 ± .001	.037 ± .001	.041 ± .002	.037 ± .001
Science	.031 ± .001	**.030 ± .001**	.032 ± .001	.050 ± .002	.032 ± .001	.031 ± .001	.035 ± .002	.031 ± .001
Enron	**.046 ± .003**	**.046 ± .003**	.049 ± .003	.176 ± .004	.053 ± .001	.061 ± .003	.055 ± .003	.047 ± .001
Yeast	.199 ± .007	.196 ± .007	.200 ± .005	.232 ± .005	**.195 ± .010**	.201 ± .009	.202 ± .006	.202 ± .008
Society	**.051 ± .002**	**.051 ± .002**	**.051 ± .001**	.063 ± .002	.053 ± .002	.052 ± .002	.055 ± .003	**.051 ± .002**
Flags	.296 ± .046	**.288 ± .017**	.299 ± .049	.573 ± .033	.335 ± .029	.371 ± .047	.346 ± .028	.336 ± .047
Birds	.052 ± .008	**.047 ± .006**	.051 ± .004	.075 ± .008	.054 ± .008	.065 ± .007	.054 ± .008	.050 ± .006
Avg rank	4.375	**2.125**	5.750	12.000	8.000	7.625	8.125	6.000

Table 3 OE results of different algorithms on 12 datasets

Data set	JLCLS	JLCLS-ELM	JLCLS-SVM	RANK-SVM	ML-KNN	LLSF	LLSF-DL	LIFT
Genbase	.002 ± .005	**.000 ± .000**	.003 ± .006	.743 ± .052	**.000 ± .000**	.003 ± .006	.002 ± .005	**.000 ± .000**
Emotions	.253 ± .077	**.229 ± .057**	.231 ± .035	.345 ± .060	.276 ± .027	.286 ± .029	.258 ± .044	.261 ± .066
Medical	.122 ± .024	**.116 ± .032**	.130 ± .030	.728 ± .052	.246 ± .037	.170 ± .028	.154 ± .048	.155 ± .039
Arts	.454 ± .017	**.437 ± .020**	.440 ± .015	.770 ± .026	.542 ± .029	.456 ± .017	.456 ± .277	.458 ± .019
Computers	**.342 ± .015**	.345 ± .011	.345 ± .023	.476 ± .028	.386 ± .023	.357 ± .020	.376 ± .016	.349 ± .029
Education	.457 ± .029	**.456 ± .013**	.461 ± .022	.685 ± .022	.492 ± .023	.465 ± .020	.494 ± .028	.471 ± .022
Science	.475 ± .019	**.467 ± .017**	.472 ± .019	.765 ± .024	.551 ± .018	.487 ± .015	.490 ± .016	.480 ± .021
Enron	**.216 ± .028**	.243 ± .046	.295 ± .029	.628 ± .053	.326 ± .038	.347 ± .025	.267 ± .031	.237 ± .031
Yeast	.227 ± .025	**.223 ± .029**	.229 ± .022	.252 ± .024	.236 ± .030	.229 ± .024	.227 ± .024	.229 ± .029
Society	.386 ± .025	.384 ± .032	**.380 ± .017**	.503 ± .023	.419 ± .018	.394 ± .019	.412 ± .030	.387 ± .018
Flags	.220 ± .112	.231 ± .089	.229 ± .106	.201 ± .068	**.200 ± .074**	.226 ± .080	.289 ± .097	.227 ± .056
Birds	.287 ± .068	.277 ± .044	**.273 ± .045**	.525 ± .082	.334 ± .063	.401 ± .039	.327 ± .043	.290 ± .037
Avg rank	3.875	**3.063**	4.875	11.250	8.625	8.250	7.688	6.375

4.3 Evaluation metrics

Five evaluation indices, which are widely used in multi-label classification, are selected to compare with the above multi-label classification algorithms, in which multi-label data set $D = \{(X_{it}, Y_{ij} | 1 \leq t \leq d, 1 \leq i \leq n, 1 \leq j \leq l)\}$, $h(\cdot)$ is a multi-label classifier, $f(\cdot,\cdot)$ is a prediction function, and rank_f is a sort function. Here are the definitions of five evaluation indicators:

Hamming loss is used to reflect the number of incorrect label classification and correct label error prediction of the evaluated object labels.

$$\text{HL}_D(h) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{|Y|}|h(x_i)\Delta Y_i|\right) \tag{26}$$

In Eq. (26), Δ refers to the symmetry difference between two sets. When Hamming loss was zero, it was the best case. The smaller $\text{HL}_D(h)$, the higher the performance of $h(\cdot)$.

One-Error is the number of times that the highest ranking of the evaluated objects is not correctly labeled:

$$\text{OE}_D(f) = \frac{1}{n}\sum_{i=1}^{n}\left[\left[\arg\max_{y \in Y} f(x_i, y)\right] \notin Y_i\right] \tag{27}$$

The best case is when the One-Error is zero. The smaller $\text{OE}_D(f)$, the higher the performance of $f(\cdot)$.

Coverage rate reflects the number of labels needed in the label sequence of the evaluated object to cover all labels.

Table 4 CV results of different algorithms on 12 datasets

Data set	JLCLS	JLCLS-ELM	JLCLS-SVM	RANK-SVM	ML-KNN	LLSF	LLSF-DL	LIFT
Genbase	**.013 ± .005**	.016 ± .012	.016 ± .011	.184 ± .028	.022 ± .017	.014 ± .005	.014 ± .008	.016 ± .012
Emotions	.297 ± .017	**.282 ± .028**	.284 ± .020	.411 ± .033	.292 ± .024	.310 ± .019	.347 ± .024	.293 ± .029
Medical	**.025 ± .009**	.027 ± .012	.037 ± .016	.170 ± .014	.055 ± .011	.048 ± .015	.055 ± .017	.040 ± .011
Arts	.214 ± .012	.206 ± .008	.174 ± .010	.254 ± .013	.187 ± .007	.219 ± .012	.270 ± .018	**.172 ± .010**
Computers	.137 ± .007	.135 ± .014	.107 ± .005	.152 ± .009	.111 ± .009	.148 ± .008	.200 ± .015	**.104 ± .009**
Education	.150 ± .014	.147 ± .008	**.101 ± .009**	.146 ± .007	.103 ± .007	.170 ± .015	.292 ± .018	.102 ± .004
Science	.168 ± .011	.175 ± .011	**.130 ± .009**	.213 ± .012	.145 ± .008	.185 ± .017	.218 ± .013	**.130 ± .008**
Enron	**.234 ± .017**	.289 ± .025	.243 ± .021	.308 ± .040	.248 ± .014	.414 ± .019	.313 ± .026	.241 ± .019
Yeast	.452 ± .014	.448 ± .016	.464 ± .008	.535 ± .019	**.447 ± .011**	.463 ± .019	.452 ± .010	.464 ± .011
Society	.231 ± .009	.235 ± .016	.191 ± .007	.253 ± .011	.192 ± .009	.244 ± .012	.290 ± .011	**.187 ± .010**
Flags	.556 ± .054	.563 ± .045	.552 ± .049	.553 ± .037	.572 ± .045	.552 ± .047	.583 ± .036	**.540 ± .039**
Birds	.161 ± .039	**.143 ± .029**	.149 ± .018	.201 ± .029	.150 ± .036	.271 ± .036	.218 ± .041	.164 ± .021
Avg rank	5.563	5.625	**4.063**	10.000	5.813	8.500	10.188	4.250

Table 5 RL results of different algorithms on 12 datasets

Data set	JLCLS	JLCLS-ELM	JLCLS-SVM	RANK-SVM	ML-KNN	LLSF	LLSF-DL	LIFT
Genbase	**.002 ± .003**	.003 ± .006	.004 ± .006	.163 ± .022	.007 ± .009	.003 ± .003	**.002 ± .004**	.004 ± .006
Emotions	.158 ± .028	.143 ± .032	**.142 ± .030**	.290 ± .030	.157 ± .017	.174 ± .020	.203 ± .019	.155 ± .029
Medical	**.015 ± .006**	.016 ± .008	.024 ± .011	.144 ± .012	.037 ± .008	.035 ± .013	.038 ± .014	.025 ± .007
Arts	.140 ± .009	.134 ± .005	.114 ± .006	.191 ± .011	.130 ± .006	.144 ± .008	.190 ± .013	**.113 ± .007**
Computers	.095 ± .005	.092 ± .009	.070 ± .004	.104 ± .006	.074 ± .005	.103 ± .007	.147 ± .011	**.067 ± .006**
Education	.101 ± .005	.100 ± .005	**.071 ± .006**	.117 ± .004	.076 ± .005	.117 ± .011	.225 ± .015	.073 ± .004
Science	.122 ± .007	.129 ± .007	.095 ± .007	.172 ± .010	.110 ± .006	.136 ± .010	.167 ± .011	**.094 ± .005**
Enron	**.081 ± .005**	.104 ± .013	.086 ± .007	.350 ± .047	.093 ± .007	.288 ± .019	.125 ± .013	.085 ± .008
Yeast	.169 ± .013	**.166 ± .016**	.174 ± .010	.235 ± .017	.167 ± .014	.174 ± .011	.172 ± .015	.173 ± .009
Society	.146 ± .008	.150 ± .013	.122 ± .005	.176 ± .009	.127 ± .006	.156 ± .008	.196 ± .007	**.119 ± .005**
Flags	.229 ± .044	.231 ± .430	.232 ± .043	.235 ± .018	.236 ± .041	.240 ± .033	.284 ± .032	**.228 ± .031**
Birds	.105 ± .026	**.091 ± .019**	.097 ± .019	.151 ± .027	.103 ± .029	.217 ± .037	.162 ± .032	.111 ± .014
Avg rank	5.063	4.938	4.250	10.938	5.875	9.188	9.813	**3.938**

Table 6 AP results of different algorithms on 12 datasets

Data set	JLCLS	JLCLS-ELM	JLCLS-SVM	RANK-SVM	ML-KNN	LLSF	LLSF-DL	LIFT
Genbase	**.997 ± .004**	.995 ± .009	.995 ± .007	.426 ± .046	.992 ± .009	.996 ± .004	.996 ± .001	.995 ± .007
Emotions	.808 ± .037	**.826 ± .035**	.821 ± .028	.684 ± .032	.802 ± .017	.790 ± .023	.784 ± .021	.805 ± .035
Medical	.912 ± .016	**.915 ± .020**	.904 ± .011	.497 ± .036	.816 ± .024	.875 ± .017	.809 ± .044	.881 ± .029
Arts	.624 ± .012	**.634 ± .010**	**.634 ± .010**	.407 ± .025	.571 ± .020	.619 ± .010	.603 ± .024	.627 ± .015
Computers	.713 ± .015	.713 ± .011	**.718 ± .013**	.596 ± .018	.686 ± .016	.702 ± .014	.670 ± .011	.712 ± .020
Education	.640 ± .022	.640 ± .012	**.647 ± .017**	.468 ± .013	.618 ± .015	.628 ± .018	.572 ± .022	.639 ± .015
Science	.614 ± .016	.614 ± .013	**.617 ± .014**	.373 ± .014	.557 ± .013	.597 ± .015	.584 ± .012	.610 ± .016
Enron	**.714 ± .014**	.695 ± .027	.679 ± .017	.565 ± .043	.624 ± .014	.550 ± .024	.660 ± .020	.689 ± .017
Yeast	.759 ± .018	**.767 ± .020**	.756 ± .013	.686 ± .018	.753 ± .018	.759 ± .011	.763 ± .015	.757 ± .012
Society	.642 ± .016	.639 ± .023	**.647 ± .010**	.524 ± .020	.626 ± .015	.633 ± .015	.606 ± .013	.646 ± .013
Flags	.802 ± .046	.801 ± .037	.797 ± .048	.801 ± .030	**.805 ± .035**	.796 ± .031	.770 ± .034	.802 ± .021
Birds	.758 ± .049	**.771 ± .035**	.768 ± .030	.627 ± .057	.728 ± .050	.630 ± .029	.706 ± .041	.743 ± .031
Avg rank	3.813	**3.563**	4.188	11.563	8.625	7.875	8.813	5.563

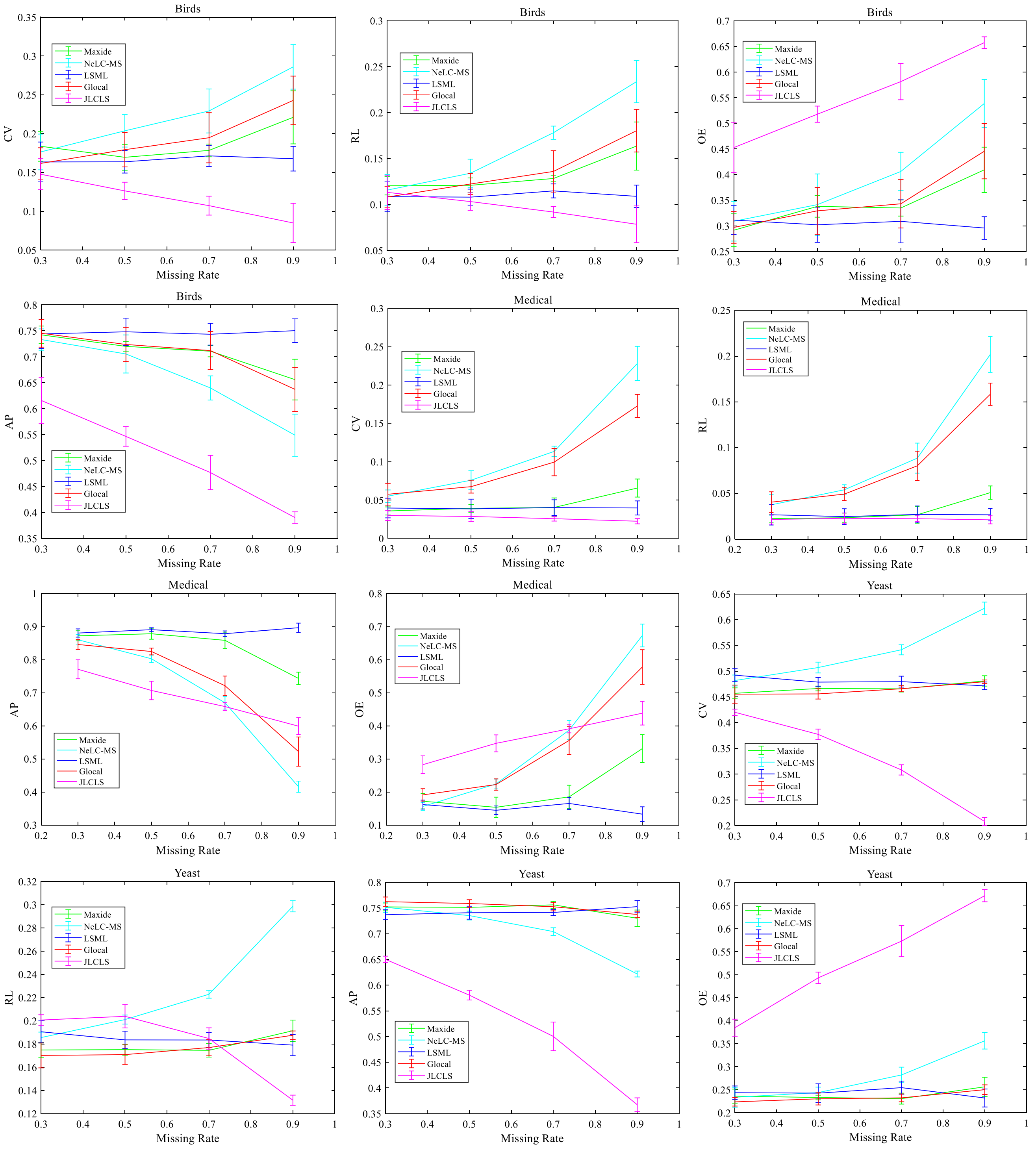


Fig. 2 Comparison results of JLCLS over birds, medical and yeast data sets against other comparing algorithms

$$\mathrm{CV}_D(f) = \frac{1}{n}\sum_{i=1}^{n}\max_{y \in Y_i} \operatorname{rank}_f(x_i, y) - 1 \tag{28}$$

The smaller the $\mathrm{CV}_D(f)$, the higher the performance of $f(\cdot,\cdot)$.

Sorting loss reflects that the ranking of non-affiliated labels is higher than the number of affiliated ones.

$$\mathrm{RL}_D(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i||\bar{Y}_i|}\cdot|\{(y_1,y_2)|f(x_i,y_1)\leq f(x_i,y_2),(y_1,y_2)\in Y_i\times\bar{Y}_i\}| \tag{29}$$

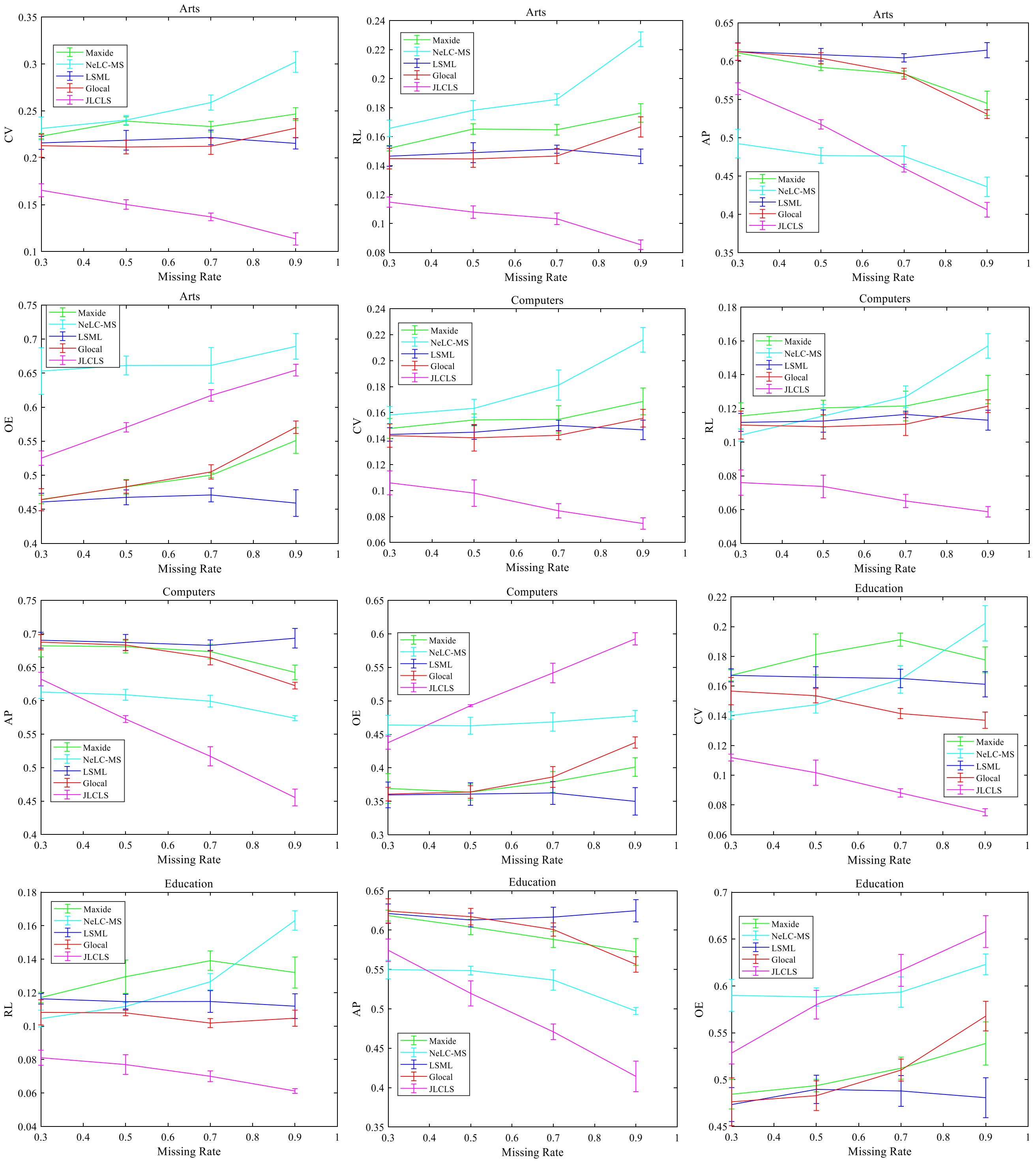


Fig. 3 Comparison results of JLCLS over arts, computers and education data sets against other comparing algorithms

The best case is when the sorting loss is zero. The smaller $\mathrm{RL}_D(f)$, the higher the performance of $f(\cdot)$.

Average accuracy is the average score of the correct labels ranked in a particular label $y \in Y_i$:

$$\mathrm{AP}_D(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|} \cdot \sum_{y \in Y_i}\frac{\left|\left\{y^{'}\middle|\mathrm{rank}_f(x_i, y^{'}) \le \mathrm{rank}_f(x_i, y), y^{'} \in Y_i\right\}\right|}{\mathrm{rank}_f(x_i, y)} \tag{30}$$

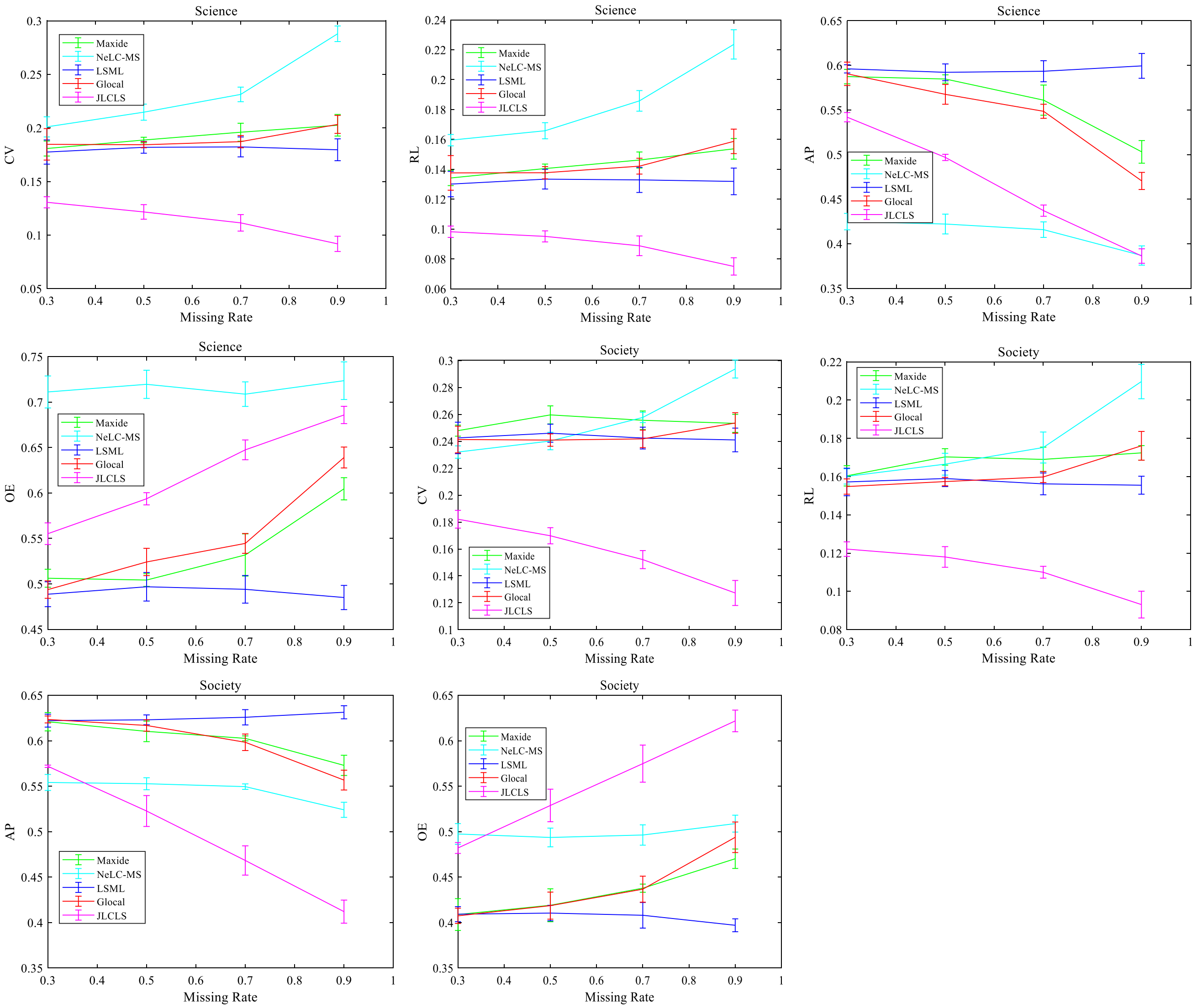


Fig. 4 Comparison results of JLCLS over arts, computers and education data sets against other comparing algorithms

The best case is when the average accurate value reaches 1. The larger $\mathrm{AP}_D(f)$, the higher the performance of $f(\cdot,\cdot)$.

4.4 Results of multi-label classification

In the experiment, the method of ten-fold cross-validation is used to evaluate the performance of the algorithm. Ten-fold cross-validation means that all data are randomly divided into ten equal subsets, each of which is tested in turn, while the rest of the data is used for training. Since ten-fold cross-validation is iterated ten times, it is necessary to calculate the average value in ten runs.

Tables 2, 3, 4, 5, and 6 show the experimental results of each algorithm under 12 data sets. In this paper, the optimal experimental results are expressed in bold. From the above experimental results, it can be observed that JLCLS-ELM ranks first in HL index on 11 data sets, and JLCLS, JLCLS-ELM, and JLCLS-SVM are slightly inferior to ML-KNN algorithm on yeast data set. In OE index, the performance of the proposed algorithm is obviously superior to that of ML-KNN on 11 data sets, and slightly inferior to that of ML-KNN on Flags data sets. In CV index, the performance of this algorithm is obviously superior in seven datasets, slightly inferior to ML-KNN in yeast dataset and slightly inferior to LIFT in four other datasets. In terms of RL index, the performance of this algorithm is obviously superior on 7 data sets, and slightly inferior to that of LIFT on the other 5 data sets. In AP index, the performance of this algorithm is obviously superior on 11 data sets, and slightly inferior to ML-KNN on yeast data sets. By comparing JLCLS-SVM with LIFT, the performance of JLCLS-SVM is predominant in HL, OE, and AP, and slightly inferior to LIFT in CV and RL evaluation. It can be seen that the algorithm proposed in this paper has a good effect.

4.5 Experimental results with missing labels

In this section, experiments are conducted on datasets with missing labels, and evaluations are performed to validate the effectiveness on both the missing labels recovery and the prediction tasks.

The dataset used in this section is part of a complete multi-label dataset. To generate missing labels, we randomly sample $\rho\%$ of the elements in the label matrix as unobserved, and the rest as missing. When $\rho = 0$, it reduces to full-label case. In this section, four evaluation metrics are selected to reflect the performance of the algorithms. For performance comparison, we consider four well-known state-of-the-art algorithms and these are the following.

1. Maxide (Xu et al. 2013) complements the label matrix by learning edge information. parameter λ^2.
2. NeLC-MS (Cheng et al. 2018) constructs a multi-label classification model by mining the hidden information in the feature space and using the correlation of labels. The smooth parameter of NeLC-MS is set to 1, and the non-equilibrium parameter is set to 0.2.
3. Glocal (Zhu et al. 2017) is a multi-label classification algorithm that processes missing labels data sets by considering global and local label correlations. The number of glocal clusters is set to 3, the regularization parameters $\lambda_2, \lambda_3, \lambda_4$ are set to $10^{-3}, 0.125, 0.125$, and the latent matrix dimension is set to 20.
4. LSML (Huang et al. 2019) considers a multi-label classification algorithm for handing missing labels data sets by learning high-order label correlation matrix and label-specific features. LSML parameter is set to $\lambda_1 = 10^2, \lambda_2 = 10^{-5}, \lambda_3 = 10^{-3}, \lambda_4 = 10^{-5}$.

Figures 2, 3, and 4 show the performance comparison of JLCLS with the other four algorithms under multi-label data with different missing rates of labels. It is easy to find that JLCLS is obviously superior to other algorithms in CV and RL evaluation metrics, but at a disadvantage in AP and OE evaluation metrics. The missing and wrong labels are the main reasons that affect the performance of multi-label algorithm. To address this problem, JLCLS first uses the original label completion matrix to complete the missing label of matrix, and then extracts a corresponding set of features for each type of label. In one instance, an incorrect label causes the label-specific features algorithm to extract this label corresponding to the wrong set of features. Similarly, a missing label causes the algorithm to fail to extract a set of features corresponding to this label. This makes label completion particularly important when the algorithm extracts label-specific features. The extracted label-specific features information needs to be used to further update and improve the label completion matrix in the process of label completion. Therefore, it is reasonable to combine label completion and label-specific features into a joint learning framework.

4.6 Visualization of label-specific features and label correlations

In Fig. 5, a and b refer to the label-specific features and label correlations obtained by JLCLS algorithm learning on emotions data sets. In Fig. 5a, the abscissa represents the index value of the label, the ordinate represents the index

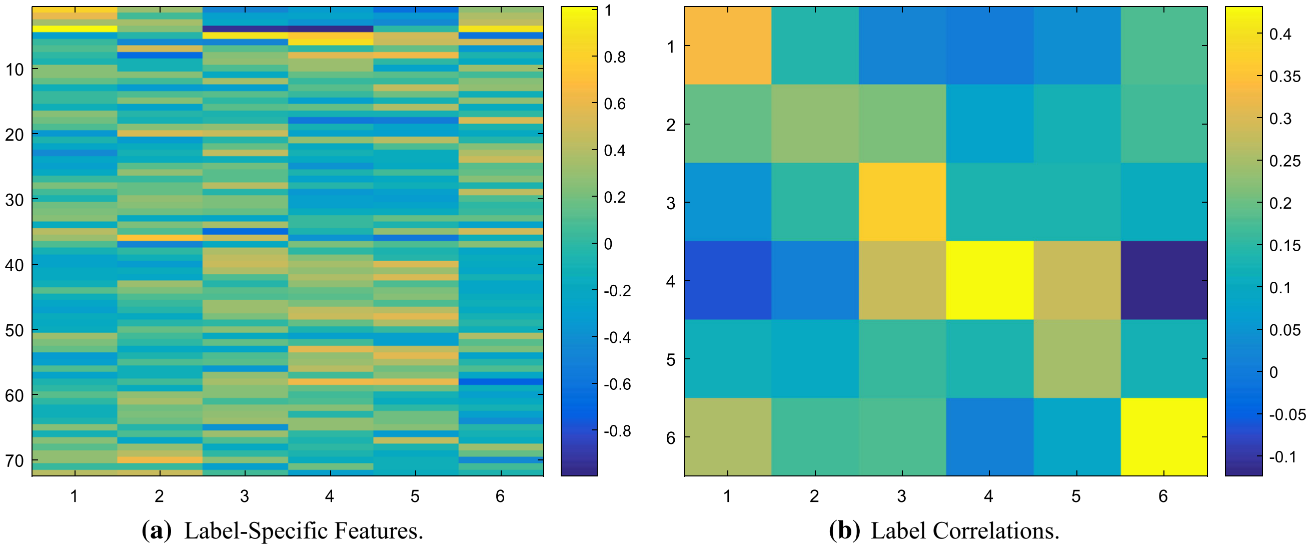


Fig. 5 Label-specific features and label correlations

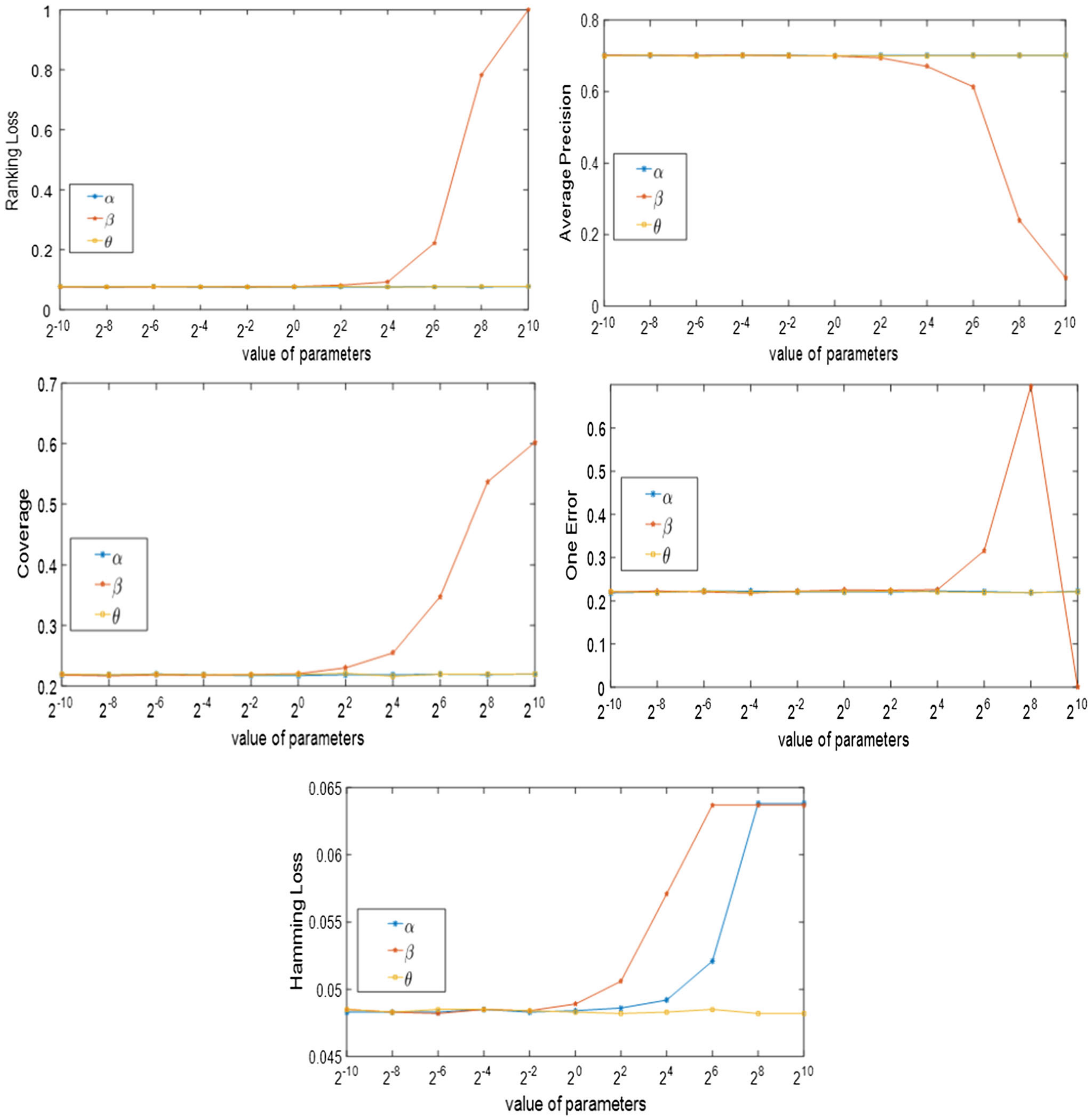
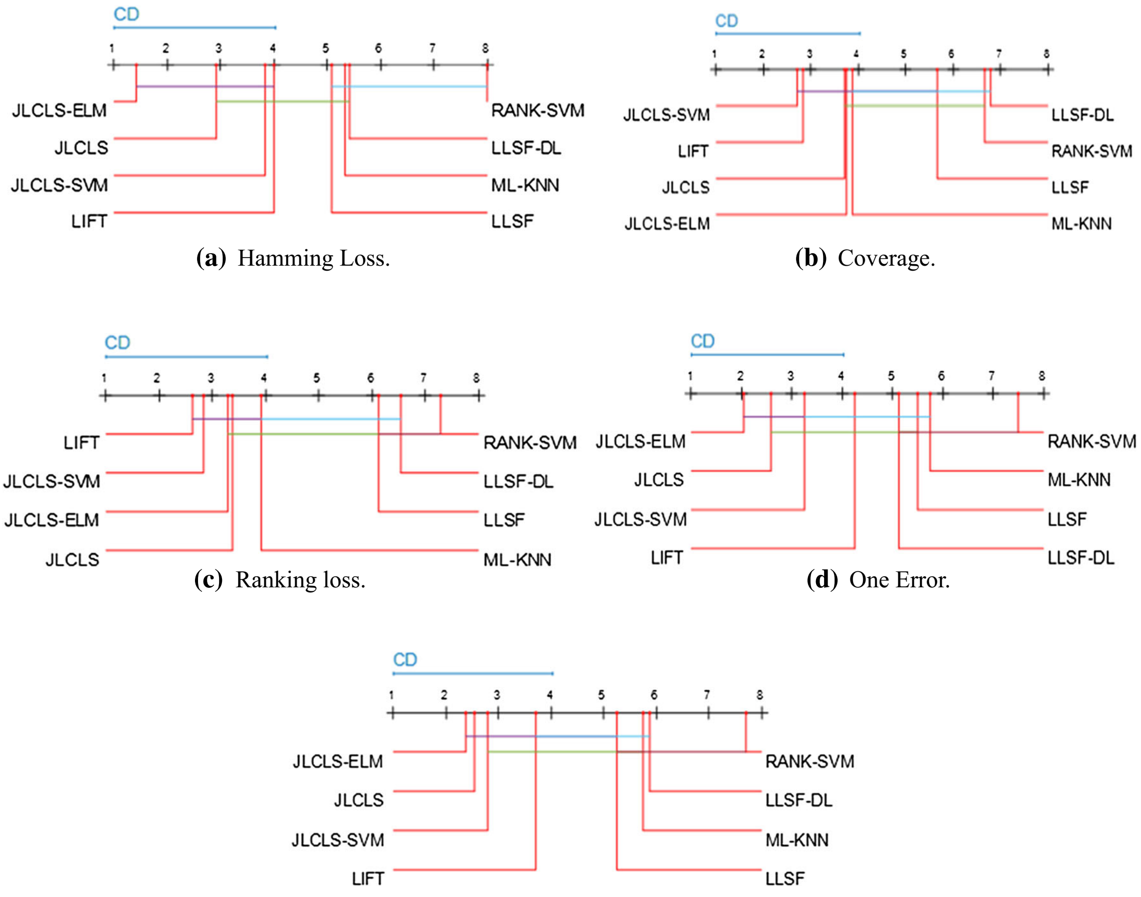


Fig. 6 Parameter sensitivity analysis

value of the feature, and the dot of the background color of the ith column represents the feature corresponding to the $\boldsymbol{y}_i$ label. Figure 5a shows the relationship between labels and features. It can be seen that each label can only extract a small number of related features from the original feature space. The sparsity of the label-specific features is affected by the parameter β. The larger the β value, the sparser the label-specific features. In Fig. 5b, both the horizontal and vertical coordinates represent the index value of the label, and the point of the ith column background color represents the correlation between the $\boldsymbol{y}_i$ labels. It can be seen that each label is associated with a small number of labels. The label correlations are also affected by the parameter β. The larger the β value, the sparser the label correlations.



(a) Hamming Loss.

(b) Coverage.

(c) Ranking loss.

(d) One Error.

(e) Average precision.

Fig. 7 Performance comparison of different algorithms

5 Parameter sensitivity analysis and hypothesis testing

5.1 Parameter sensitivity analysis

In Fig. 6, there are three parameters α, β, θ. Parameter α controls the influence of label correlations on model coefficients, parameter β controls the sparsity of label-specific features, and parameter θ controls the sparsity of difference matrix. In order to analyze the parameter sensitivity of JLCLS, the experiment of parameter sensitivity is carried out on enron data set. In this paper, the method of fixing two parameter values and changing one parameter value is used to find the optimal value. From the experimental results reflected in Fig. 6, it can be found that the performance of JLCLS algorithm is relatively insensitive to regularization parameters. In Fig. 6, we can find the appropriate parameters to optimize the performance of JLCLS algorithm. When the values of the parameters α and β are very large, the performance of the JLCLS algorithm becomes very poor. When the value of the parameter α becomes large, the JLCLS algorithm obtains information on the label correlations becoming very sparse. When the value of the parameter β is large, the features that the label-specific features can extract in the JLCLS algorithm become very sparse.

5.2 Statistical hypothesis test

In this paper, the performance stability of JLCLS, JLCLS-ELM, JLCLS-SVM and other comparative experimental algorithms on 12 data sets was compared by using Nemenyi (Demšar 2006) with 5% significance level. When the difference of the average ranking of the two comparison algorithms on all data sets is greater than the critical difference (CD), it is considered that there is a significant difference between the two algorithms, otherwise there is nothing to be considered. The formula for calculating the CD value is:

$$\mathrm{CD} = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{31}$$

In Eq. (31), $q_\alpha = 3.0310, k = 8, N = 12$, so the CD value is 3.0310. In Fig. 7, it can be found that JLCLS-ELM is significantly different from RANK-SVM, LLSF-DL, ML-KNN, and LLSF in terms of HL index, and JLCLS and RANK-SVM are significantly different. In CV index, JLCLS-SVM is significantly different from LLSF-DL, RANK-SVM, and LLSF. In RL index, although LIFT is significantly different from LLSF, LLSF-DL, and RANK-SVM, the algorithm of this paper has no significant difference with LIFT. In OE index, JLCLS-ELM is significantly different from other comparison algorithms. JLCLS is significantly different from LLSF, ML-KNN, and RANK-SVM. JLCLS-SVM and RANK-SVM are significantly different. In AP index, JLCLS and JLCLS-ELM are significantly different from LLSF, LLSF-DL, ML-KNN, and RANK-SVM, respectively. JLCLS-SVM is significantly different from LLSF-DL, ML-KNN, and RANK-SVM. It is not difficult to see from Fig. 7 that the proposed algorithm ranks first in four evaluation indexes. Although the stability of RL index is slightly lower than that of LIFT algorithm, it has no significant difference with LIFT algorithm and is superior to other algorithms. The statistical hypothesis test further illustrates the effectiveness of the proposed algorithm, the rationality of joint label completion and the learning of label-specific features.

6 Conclusion

In this paper, a new multi-label joint learning algorithm JLCLS is proposed to improve the performance of multi-label algorithm by means of joint label completion and label-specific features. Specifically, it proposes an optimization framework that combines label completion and label-specific features. In the optimization process, the label correlations are taken into account. At the same time, class attributes can be learned by multi-label classifier ELM and BSVM. In the experimental analysis, by comparing the 12 multi-label data sets with other advanced multi-label learning algorithms, we can see that the proposed algorithm has certain advantages. This algorithm can further consider classifying on default multi-label data sets. In the future, we will further study the combination of local and global label relations and label-specific features.

Compliance with ethical standards

Conflict of interest The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

Beck A, Teboulle M (2009a) Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Trans Image Process 18(11):2419–2434

Beck A, Teboulle M (2009b) A fast iterative shrinkage–thresholding algorithm for linear inverse problems. SIAM J Imaging Sci 2(1):183–202

Chen M, Zheng A, Weinberger K (2013) Fast image tagging. In: International conference on machine learning, pp 1274–1282

Cheng YS, Zhao DW, Zhan WF et al (2018) Multi-label learning of non-equilibrium labels completion with mean shift. Neurocomputing 321:92–102

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Dong HC, Li YF, Zhou ZH (2018) Learning from semi-supervised weak-label data. In: Thirty-second AAAI conference on artificial intelligence, pp 2926–2933

Elisseeff A, Weston J (2002) A kernel method for multi-labelled classification. In: Advances in neural information processing systems, pp 681–687

Fu B, Xu G, Wang Z, et al (2013) Leveraging supervised label dependency propagation for multi-label learning. In: 2013 IEEE 13th international conference on data mining. IEEE, pp 1061–1066

Guo K, Cao R, Kui X et al (2019) LCC: towards efficient label completion and correction for supervised medical image learning in smart diagnosis. J Netw Comput Appl 133:51–59

Han H, Huang M, Zhang Y et al (2019) Multi-label learning with label specific features using correlation information. IEEE Access 7:11474–11484

He ZF, Yang M, Gao Y et al (2019) Joint multi-label classification and label correlations with missing labels and feature selection. Knowl Based Syst 163:145–158

Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501

Huang J, Li G, Huang Q, et al (2015) Learning label specific features for multi-label classification. In: 2015 IEEE international conference on data mining. IEEE, pp 181–190

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

Huang J, Li G, Huang Q et al (2018) Joint feature selection and classification for multilabel learning. IEEE Trans Cybern 48(3):876–889

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

Liu G, Lin Z, Yu Y (2010) Robust subspace segmentation by low-rank representation. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp 663–670

Liu B, Li Y, Xu Z (2018) Manifold regularized matrix completion for multi-label learning with ADMM. Neural Netw 101:57–67

Nguyen TT, Nguyen TTT, Luong AV et al (2019) Multi-label classification via label correlation and first order feature dependance in a data stream. Pattern Recogn 90:35–51

Rodriguez P, Wohlberg B (2016) Incremental principal component pursuit for video background modeling. J Math Imaging Vis 55(1):1–8

Sitompul OS, Nababan EB (2018) Biased support vector machine and weighted-smote in handling class imbalance problem. Int J Adv Intell Inform 4(1):21–27

Sun YY, Zhang Y, Zhou ZH (2010) Multi-label learning with weak label. In: Twenty-fourth AAAI conference on artificial intelligence, pp 593–598

Wu B, Lyu S, Ghanem B (2015) ML–MG: multi-label learning with missing labels using a mixed graph. In: Proceedings of the IEEE international conference on computer vision, pp 4157–4165

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

Xu L, Wang Z, Shen Z, et al (2014) Learning low-rank label correlations for multi-label classification with missing labels. In: 2014 IEEE international conference on data mining. IEEE, pp 1067–1072

Xu S, Yang X, Yu H et al (2016) Multi-label learning with label-specific feature reduction. Knowl Based Syst 104:52–61

Zendehboudi A, Baseer MA, Saidur R (2018) Application of support vector machine models for forecasting solar and wind energy resources: a review. J Clean Prod 199:272–285

Zhang ML, Wu L (2015) Lift: multi-label learning with label-specific features. IEEE Trans Pattern Anal Mach Intell 37(1):107–120

Zhang ML, Zhou ZH (2007) ML-KNN: a lazy learning approach to multi-label learning. Pattern Recogn 40(7):2038–2048

Zhang J, Li C, Cao D et al (2018) Multi-label learning with label-specific features by resolving label correlations. Knowl Based Syst 159:148–157

Zhao D, Wang T, Chu F (2019) Deep convolutional neural network based planet bearing fault classification. Comput Ind 107:59–66

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