Applied Intelligence
https://doi.org/10.1007/s10489-021-02725-4

Asymmetry label correlation for multi-label learning

Jiachao Bao [1,2] · Yibin Wang [1,3] · Yusheng Cheng [1,3]

Accepted: 27 July 2021

Abstract
As an effective method for mining latent information between labels, label correlation is widely adopted by many scholars to model multi-label learning algorithms. Most existing multi-label algorithms usually ignore that the correlation between labels may be asymmetric while asymmetry correlation commonly exists in the real-world scenario. To tackle this problem, a multi-label learning algorithm with asymmetry label correlation (ACML, Asymmetry Label Correlation for Multi-Label Learning) is proposed in this paper. First, measure the adjacency between labels to construct the label adjacency matrix. Then, cosine similarity is utilized to construct the label correlation matrix. Finally, we constrain the label correlation matrix with the label adjacency matrix. Thus, asymmetry label correlation is modeled for multi-label learning. Experiments on multiple multi-label benchmark datasets show that the ACML algorithm has certain advantages over other comparison algorithms. The results of statistical hypothesis testing further illustrate the effectiveness of the proposed algorithm.

Keywords Multi-label classification · Asymmetry label correlation · Label correlation

1 Introduction

Multi-label learning [1, 2] associates an instance with multiple labels, which is more adaptive for processing rich semantics objects in the real-world. However, as the data volume continue to increase, the output space of the algorithm also increases exponentially. Higher dimensionality results in difficulty to mine the mapping relationship between features and labels. The higher dimensionality of the feature space and the label space affects the performance of the multi-label classification algorithm to a certain extent. In multi-label learning, labels are usually correlated with each other. Except for the information provided by the high dimensional features, such correlation can also provide extra vital information for multi-label learning algorithms. In order to improve multi-label learning with high-dimensional data more effectively, scholars proposed label correlation [3] to mine the latent information in the label space.

In order to mine the latent information in the label space. Xue [4] et al. proposed a multi-label classification algorithm CC Net (Learning semantic dependencies with channel correlation for multi-label classification) based on the attention mechanism and convolutional neural network. By adding an attention module to the convolutional neural network, channel features for each label and correlation between pairwise label are learned respectively. The LSF-CI algorithm (multi-label learning with label-specific features using correlation information) proposed by Han [5] et al. assumes that labels are associated with their own unique features. Features in the original feature space contribute varies to different labels, and similar labels have similar features. By considering the correlation among labels, a sparse feature coefficient matrix for extracting label-specific features is learned. Zhu [6] et al. proposed the multi-label learning method GLOCAL (Multi-Label Learning with Global and Local Label Correlation) based on global and local label correlation. The GLOCAL algorithm obtains latent labels through the low-rank decomposition of the label matrix. The mapping between latent label space and feature space and the global and local label correlations are learned jointly. Cheng [7] et al. proposed the FF-IMLL algorithm (multi-label lazy learning approach based on firefly method)

✉ Yusheng Cheng
chengyshaq@163.com

Jiachao Bao
905815061@qq.com

Yibin Wang
wangyb07@mail.ustc.edu.cn

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

[2] School of Computer and Software Engineering, Anhui Institute of Information Technology, Wuhu 241000, Anhui, China

[3] Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou 363000, Fujian, China

Published online: 31 August 2021

achieved certain performance improvements by considering both nearest neighbor and label similarity simultaneously.

In the research of multi-label learning, many scholars have introduced label correlation into the model. Label correlation considered to be an important means of mining latent information in feature space and label space. Label correlation assumes that labels with a higher correlation have a higher probability to appearing in the same instance. However, most existing label correlation method only considers the degree of correlation while ignores the correlate direction in the correlation relationship. That is, most existing methods only consider the symmetric label correlation. Only considering the symmetric label correlation will introduce redundant information in the model, resulting in classifier performance decreasing. In real life, asymmetric correlation commonly exists. For instance, the reading of the thermometer is directly related to the room temperature, but manually changing the thermometer's reading will not affect the actual room temperature, the correlation between the thermometer's reading and the actual room temperature is asymmetric. In multi-label learning, the correlation between labels can also be asymmetry [8, 9]. Simply considering symmetric correlation cannot modeling certain scenario in real-world data where asymmetric correlation exits. To deal with this problem, a multi-label learning algorithm with asymmetry label correlation is proposed in this paper, the proposed algorithm measure the adjacency relationship between labels to consider the asymmetry in label correlation.

To consider the asymmetry of label correlation in multi-label learning, a multi-label learning algorithm ACML (Asymmetry Label Correlation for Multi-Label Learning) based on distance correlation [10] and conditional independence test [11] is proposed in this paper. The basic idea is to measure the dependency from both possible direction and select the smaller output as the inferred direction. Specifically, we first use distance correlation to infer the adjacency relation between labels and construct the label adjacency matrix. Meanwhile, the label correlation matrix is obtained by cosine similarity. Second, the label adjacency matrix and the label correlation matrix are employed to construct the asymmetric label correlation matrix. The asymmetric label correlation matrix is used to learn the multi-label classification model. Finally, the proposed algorithm is evaluated with other multi-label classification algorithms on 13 benchmark multi-label datasets. The experimental results show that the introduction of asymmetric label correlation in the model can improve the performance of the algorithm to a certain extent. (Our codes: https://github.com/chengyshaq/code-for-CCML).

The rest of this paper is organized as follows. In section 2, we introduce the basic ideal of the DC method and propose a multi-label learning algorithm based on the DC method. Section 3, the pseudocode, and complexity analysis of the proposed algorithm is given. Section 4, datasets, metrics, and parameters setup are introduced. Section 5, experimental results on 13 benchmark datasets are given, we validate the proposed method with a hypothesis test. Finally, we conclude this paper in section 6.

2 The proposed method

2.1 Label adjacency matrix

In graph theory [12], the adjacency between nodes usually represents by a square matrix G named adjacency matrix. Each element of the adjacency matrix indicates whether the corresponding node pairs are adjacent or not. Specially, the adjacency matrix for a directed acyclic graph (DAG) can be asymmetric [13, 14], where each element suggests whether the nodes are adjacent on that specific direction. In this paper, we utilize a square (0, 1)-matrix with zeros on its diagonal as the label adjacency matrix, where 0 suggests nodes are not adjacent on that orientation, 1 indicates nodes are adjacent on that direction. To learn the label adjacency matrix, we adopted the DC method [15]. The DC method performs conditional independence test on variables constructed from the original label data. The distance correlation is utilized to measure the dependency of both possible direction between the constructed variables, direction with smaller distance correlation is inferred as the adjacency orientation.

Given variables $\{X, Y\}$, to infer the adjacency direction between $\{X, Y\}$, DC method consider $(P(X), P(Y|X))$ as a realization of a variable pair and calculates the distance correlation $\mathcal{D}(P(Y), P(X|Y))$. The distance correlation is a measurement for dependency, the bigger the distance correlation is, the more likely $P(X)$ and $P(Y|X)$ are independent on each other. For adjacency direction inference scenario, distance correlation of $\mathcal{D}(P(X), P(Y|X))$ and $\mathcal{D}(P(Y), P(X|Y))$ are calculated. The decision is made based on a given threshold ε, if $\mathcal{D}(P(X), P(Y|X)) - \mathcal{D}(P(Y), P(X|Y)) > \varepsilon$, then the adjacency direction is $Y \rightarrow X$, vice versa. According to Liu [15] et al. in most case, the DC method achieves the best results when $\varepsilon = 0$.

For a multi-label dataset, $\boldsymbol{Y} \in \mathbb{R}^{n \times l}$ is the label matrix, suppose y_i, y_j as a pair of labels, where $i, j = 1, 2, \cdots, n$. $P(y_i)$ is the distribution of the label y_i. $P\left(y_i^0\right)$ is the probability of $y_i = 0$, $P\left(y_i^1\right)$ is the probability of $y_i = 1$, $P(y_i, y_j)$ is the joint distribution of label y_i, and label y_j, $P(y_j|y_i)$ is the conditional distribution of y_j given y_i, $P\left(y_j|y_i^0\right)$ is the conditional distribution of y_j given $y_i = 0$, $P\left(y_j|y_i^1\right)$ is the conditional distribution of y_j given $y_i = 1$. The DC method treats $P(y_i)$ and $P(y_j|y_i)$ as two independent random variables. $P\left(y_i^0\right)$, $P\left(y_i^1\right)$ are one-dimensional random variables, while $P\left(y_j|y_i^0\right)$, $P\left(y_j|y_i^1\right)$ are high-dimensional random variables. As shown in Fig. 1, the DC method measuring the distance correlation of $y_i \rightarrow y_j$ and $y_j \rightarrow y_i$. Intuitively, the

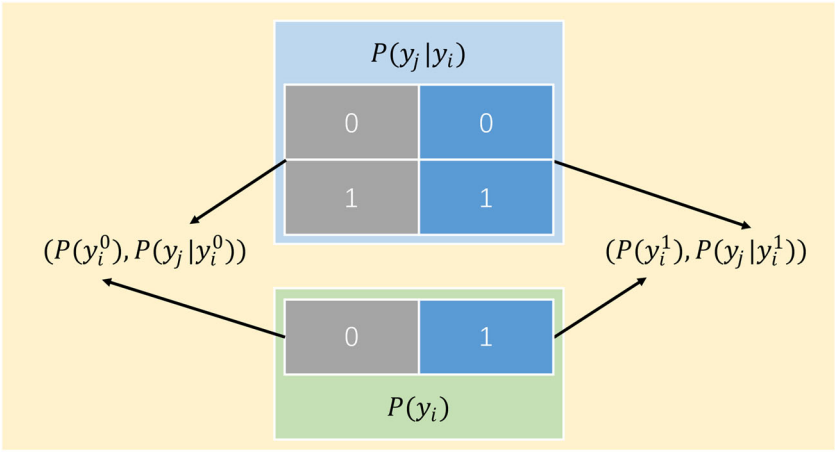


Fig. 1 $P(y_i)$ and $P(y_j|\ y_i)$ as a random variable pair

direction that has a smaller dependence coefficient is the inferred adjacency direction, as the variables are more dependent.

The steps of inferring adjacency direction on label data using the DC method are as follows:

Let $\lambda = P(y_i)$, $\mu = P(y_j|\ y_i)$, suppose $(\lambda,\ \mu)$ is a pair of independent random variables, f_λ and f_μ correspond to their characteristic function, respectively. $f_{\lambda,\ \mu}$ is their joint characteristic function, then the distance covariance of $(\lambda,\ \mu)$ is defined as:

$$\mathcal{C}^2(\lambda,\mu) = \left\|f_{\lambda,\mu} - f_\lambda f_\mu\right\|^2 \tag{1}$$

Then the distance correlation $\mathcal{D}(\lambda,\mu)$ is:

$$\mathcal{D}(\lambda,\mu) = \frac{\mathcal{C}(\lambda,\mu)}{\sqrt{\mathcal{C}(\lambda,\lambda)\mathcal{C}(\mu,\mu)}} \tag{2}$$

And if $\mathcal{C}(\lambda,\lambda) = 0$ or $\mathcal{C}(\mu,\mu) = 0$, then $\mathcal{D}(\lambda,\mu) = 0$.

Suppose the multi-label dataset contains n instances, and l labels, then for any pair of labels $(X_i,\ Y_j\,|\,i,\ j = 1,\ 2,\ 3,\ \ldots,\ l)$ can be construct n groups of variables $\{(\lambda_i\ ,\mu_j)\}_{i,j=1}^n$. For variables λ and μ, construct the following [10, 15]:

$$a_{ij} = \left\|\lambda_i - \lambda_j\right\| \quad a_{i\cdot} = \frac{1}{n}\sum_{j=1}^{n}\lambda_{ij} \quad a_{\cdot j} = \frac{1}{n}\sum_{i=1}^{n}\lambda_{ij} \quad a_{\cdot\cdot} = \frac{1}{n^2}\sum_{i,j=1}^{n}\lambda_{ij} \tag{3}$$

$$b_{ij} = \left\|\mu_i - \mu_j\right\| \quad b_{i\cdot} = \frac{1}{n}\sum_{j=1}^{n}\mu_{ij} \quad b_{\cdot j} = \frac{1}{n}\sum_{i=1}^{n}\mu_{ij} \quad b_{\cdot\cdot} = \frac{1}{n^2}\sum_{i,j=1}^{n}\mu_{ij} \tag{4}$$

Furtherly, we can construct matrix $\boldsymbol{A}$, $\boldsymbol{B}$ according to the definition of distance correlation [10, 15]:

$$\boldsymbol{A}_{ij} = a_{ij} - a_{i\cdot} - a_{\cdot j} + a_{\cdot\cdot} \tag{5}$$

$$\boldsymbol{B}_{ij} = b_{ij} - b_{i\cdot} - b_{\cdot j} + b_{\cdot\cdot} \tag{6}$$

Then the distance covariance can be calculated as follows [15]:

$$\mathcal{C}_n(\lambda,\mu) = \frac{1}{n}\sqrt{\sum_{i,j=1}^{n}\boldsymbol{A}_{ij}\boldsymbol{B}_{ij}} \tag{7}$$

According to Eq. (2) and Eq. (7), we can further write:

$$\mathcal{D}(\lambda,\mu) = n\frac{\sqrt{\sum_{i,j=1}^{n}\boldsymbol{A}_{ij}\boldsymbol{B}_{ij}}}{\sqrt{\sum_{i,j=1}^{n}\boldsymbol{A}_{ij}\boldsymbol{A}_{ij}}\sqrt{\sum_{i,j=1}^{n}\boldsymbol{B}_{ij}\boldsymbol{B}_{ij}}} \tag{8}$$

For any pair of labels $(y_i,\ y_j\,|\,i,\ j = 1,\ 2,\ 3,\ \ldots,\ l)$:

If $\mathcal{D}_{y_j\to y_i} > \mathcal{D}_{y_i\to y_j}$, then inferred adjacency direction is $y_i \to y_j$,

If $\mathcal{D}_{y_i\to y_j} > \mathcal{D}_{y_j\to y_i}$, then inferred adjacency direction is $y_j \to y_i$.

From above, we can construct label adjacency matrix $\boldsymbol{C} \in \{0,\ 1\}^{l\ \times\ l}$, where $\boldsymbol{C}_{ij} = 0$ indicates the i-th label and the j-th label are not adjacent, while $\boldsymbol{C}_{ij} = 1$ indicates the inferred adjacency direction is $i \to j$.

2.2 The proposed method

In multi-label learning [1, 2], Let $\boldsymbol{X} = \{\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots,\ \boldsymbol{x}_n\}$ be the training data, $\boldsymbol{Y} = \{\boldsymbol{y}_1,\ \boldsymbol{y}_2,\ \ldots,\ \boldsymbol{y}_n\}$ be the corresponding label matrix, where $\boldsymbol{X} \in \mathbb{R}^{n\ \times\ d}$, $\boldsymbol{Y} \in \mathbb{R}^{n\ \times\ l}$, n is the number of instances, d is the number of features, l is the number of labels. Then $D = \{(\boldsymbol{x}_i,\ \boldsymbol{y}_i)\ |\ i = 1, 2, \cdots, n\}$ is a multi-label dataset, where $\boldsymbol{x}_j = \left\{\boldsymbol{x}_i^1\ ,\boldsymbol{x}_i^2, \cdots, \boldsymbol{x}_i^d\right\}$ is a feature vector, $\boldsymbol{y}_i = \left\{\boldsymbol{y}_i^1\ ,\boldsymbol{y}_i^2\ , \cdots, \boldsymbol{y}_i^l\right\}$ is a label vector. Then, the task of multi-label learning is to find a mapping between the training set $\boldsymbol{X}$ and the label matrix $\boldsymbol{Y}$, that is, $f : \boldsymbol{X} \to 2^{\boldsymbol{Y}}$. The basic model of multi-label learning can be written as:

$$\min_{\boldsymbol{W}} \|\boldsymbol{X}\boldsymbol{W} - \boldsymbol{Y}\|_F^2 + \alpha\|\boldsymbol{W}\|_1 \tag{9}$$

where $\boldsymbol{W} \in \mathbb{R}^{d\ \times\ l}$ is the coefficient matrix, α is the sparse parameter that controls the sparsity of matrix $\boldsymbol{W}$. According to

Zhang [1] et al. in multi-label learning, labels are usually correlated to each other. That is, correlated labels are more likely to appear in the same instance at same time. Consider label correlation can further improve the performance of the multi-label learning. By adding the label correlation into the model, the model can further write as:

$$\min_{\boldsymbol{W}} \|\boldsymbol{XW}-\boldsymbol{Y}\|_F^2 + \frac{\beta}{2} tr\left(\boldsymbol{RW}^{\mathrm{T}}\boldsymbol{W}\right) + \alpha\|\boldsymbol{W}\|_1 \tag{10}$$

where $\boldsymbol{R} \in \mathbb{R}^{l \times l}$ is the label correlation matrix, β is a model parameter that controls the impact of label correlation matrix. This paper uses cosine similarity to calculate label correlation.

Some existing multi-label learning algorithms usually directly add the label correlation matrix to the model which only considering symmetric label correlation, ignoring the asymmetry relationship between the labels. The algorithm proposed in this paper uses the label correlation matrix and the label adjacency matrix to construct the asymmetric label correlation matrix to consider the asymmetry relationship between labels.

Definition 1 $\boldsymbol{V}$ is the asymmetric label correlation matrix, $\boldsymbol{V}$ is a square matrix with dimension of $l \times l$. If $\boldsymbol{V_{ij}} = \boldsymbol{V_{ji}}$, the correlation between label i and label j is symmetric. If $\boldsymbol{V_{ij}} = \boldsymbol{0}, \boldsymbol{V_{ji}} = \boldsymbol{1}$ or $\boldsymbol{V_{ij}} = \boldsymbol{1}, \boldsymbol{V_{ji}} = \boldsymbol{0}$, the correlation between label i and label j is asymmetric.

Definition 2 $\boldsymbol{R} \in \mathbb{R}^{l \times l}$ is the label correlation matrix, $\boldsymbol{C} \in \{0, 1\}^{l \times l}$ is the label adjacency matrix. $\boldsymbol{C}^{\mathrm{C}}$ is the complement matrix of $\boldsymbol{C}$. Then the asymmetric label correlation matrix $\boldsymbol{V}$ is:

$$\boldsymbol{V} = \boldsymbol{RC}^{\mathrm{C}} \tag{11}$$

The obtained asymmetric label correlation matrix $\boldsymbol{V}$ is added to the basic model, which helps to remove the redundancy parameters from the model. According to Han [5] et al. reduce model size force the algorithm to learn a more general model, which in turns improve the performance. Then the Eq. (10) can be further written as:

$$\min_{\boldsymbol{W}} \|\boldsymbol{XW}-\boldsymbol{Y}\|_F^2 + \frac{\beta}{2} tr\left(\boldsymbol{VW}^{\mathrm{T}}\boldsymbol{W}\right) + \alpha\|\boldsymbol{W}\|_1 \tag{12}$$

2.3 Model optimization

In following part, the optimization of the model is given. Since the learning objective $\boldsymbol{W}$ constrained by a l_1 norm, the $\boldsymbol{W}$ is non-smooth and cannot be solved directly by derivation. Therefore, this paper uses accelerated gradient descent to solve the model parameter $\boldsymbol{W}$. Beck [16] et al. pointed out that for the following convex function optimization problem:

$$\min_{\boldsymbol{W}\in\mathcal{H}} F(\boldsymbol{W}) = f(\boldsymbol{W}) + \mathrm{g}(\boldsymbol{W}) \tag{13}$$

Where $\mathcal{H}$ is Hilbert space, $f(\boldsymbol{W})$ and $g(\boldsymbol{W})$ are convex functions, $f(\boldsymbol{W})$ satisfies the Lipschitz condition, then for any matrix $\boldsymbol{W_1}$ and $\boldsymbol{W}_2$ we have:

$$\|\nabla f(\boldsymbol{W}_1) - \nabla f(\boldsymbol{W}_2)\| \le L_g\|\Delta\boldsymbol{W}\| \tag{14}$$

Where L_g is the Lipschitz constant, and $\Delta\boldsymbol{W} = \boldsymbol{W}_1 - \boldsymbol{W}_2$. In accelerated gradient descent, instead of directly minimize $F(\boldsymbol{W})$, we introduce $Q(\boldsymbol{W}, \boldsymbol{W}^{(t)})$ to approximate $F(\boldsymbol{W})$, $Q(\boldsymbol{W}, \boldsymbol{W}^{(t)})$ is defined as:

$$Q\left(\boldsymbol{W}, \boldsymbol{W}^{(t)}\right) = f\left(\boldsymbol{W}^{(t)}\right) + \left\langle \nabla f\left(\boldsymbol{W}^{(t)}\right), \boldsymbol{W}-\boldsymbol{W}^{(t)} \right\rangle + \frac{L_g}{2}\left\|\boldsymbol{W}-\boldsymbol{W}^{(t)}\right\|_F^2 + g(\boldsymbol{W}) \tag{15}$$

Let

$$G^{(t)} = \boldsymbol{W}^{(t)} - \frac{1}{L_g}\nabla f\left(\boldsymbol{W}^{(t)}\right) \tag{16}$$

Then Eq. (15) can be further written as:

$$\boldsymbol{W} = \arg\min_{\boldsymbol{W}} g(\boldsymbol{W}) + \frac{L_g}{2}\left\|\boldsymbol{W}-G^{(t)}\right\|_F^2 \tag{17}$$

According to Eq. (12):

$$f(\boldsymbol{W}) = \|\boldsymbol{XW}-\boldsymbol{Y}\|_F^2 + \frac{\beta}{2} tr\left(\boldsymbol{VW}^{T}\boldsymbol{W}\right) \tag{18}$$

$$g(\boldsymbol{W}) = \alpha\|\boldsymbol{W}\|_1 \tag{19}$$

Then:

$$\boldsymbol{W} = \arg\min_{\boldsymbol{W}} \frac{1}{2}\left\|\boldsymbol{W}-G^{(t)}\right\|_F^2 + \frac{\alpha}{L_g}\|\boldsymbol{W}\|_1 \tag{20}$$

According to Ganesh [17] et al. Let:

$$\boldsymbol{W}^{(t)} = \boldsymbol{W}_t + \frac{b_{t-1}-1}{b_t}(\boldsymbol{W}_t - \boldsymbol{W}_{t-1}) \tag{21}$$

When b_t satisfies $b_{t+1}^2 - b_{t+1} \le b_t^2$, the convergence speed of the algorithm can be improved to $O(t^{-2})$, $\boldsymbol{W}_t$ can be regarded as the result of the t-th iteration of $\boldsymbol{W}$.

In this paper $g(\boldsymbol{W})$ is a l_1 norm. In each iteration, $\boldsymbol{W}$ can be solved as following problem:

$$\boldsymbol{W}_{t+1} = S_\varepsilon\left[G^{(t)}\right] = \arg\min_{\boldsymbol{W}} \varepsilon\|\boldsymbol{W}\|_1 + \frac{1}{2}\left\|\boldsymbol{W}-G^{(t)}\right\|_F^2 \tag{22}$$

Where $S_\varepsilon[\cdot]$ is the soft threshold operator, for each $\boldsymbol{w}_{ij}$ and $\varepsilon = \frac{\alpha}{L_g} > 0$, the soft threshold operator is defined as:

$$S_\varepsilon(w_{ij}) = \begin{cases} \boldsymbol{w}_{ij} - \varepsilon & \text{when } \boldsymbol{w}_{ij} > \varepsilon \\ \boldsymbol{w}_{ij} + \varepsilon & \text{when } \boldsymbol{w}_{ij} < -\varepsilon \\ 0 & \text{others} \end{cases} \tag{23}$$

We continue to prove the Lipschitz continuity of Eq. (12). Since the asymmetric label correlation matrix is an asymmetric matrix, the calculation of Lipschitz constant is slightly different. According to Eq. (12), $\nabla f(\boldsymbol{W})$ is:

$$\nabla f(\boldsymbol{W}) = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W} - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y} + \frac{\beta}{2}\left(\boldsymbol{W}\boldsymbol{V} + \boldsymbol{W}\boldsymbol{V}^{\mathrm{T}}\right) \tag{24}$$

For a given $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$:

$$\begin{aligned}
&\|\nabla f(\boldsymbol{W}_1) - \nabla f(\boldsymbol{W}_2)\|_F^2 = \left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W} + \frac{\beta}{2}\left(\boldsymbol{W}\boldsymbol{V} + \boldsymbol{W}\boldsymbol{V}^{\mathrm{T}}\right)\right\|_F^2 \\
&\le 2\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}\right\|_F^2 + 2\left\|\frac{\beta}{2}\boldsymbol{W}\boldsymbol{V}\right\|_F^2 + 2\left\|\frac{\beta}{2}\boldsymbol{W}\boldsymbol{V}^{\mathrm{T}}\right\|_F^2 \\
&\le 2\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right\|_2^2\|\Delta\boldsymbol{W}\|_F^2 + \frac{1}{2}\|\beta\boldsymbol{V}\|_2^2\|\Delta\boldsymbol{W}\|_F^2 + \frac{1}{2}\left\|\beta\boldsymbol{V}^{\mathrm{T}}\right\|_2^2\|\Delta\boldsymbol{W}\|_F^2 \\
&= \left(2\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right\|_2^2 + \frac{1}{2}\|\beta\boldsymbol{V}\|_2^2 + \frac{1}{2}\left\|\beta\boldsymbol{V}^{\mathrm{T}}\right\|_2^2\right)\|\Delta\boldsymbol{W}\|_F^2 \\
&= \left(2\sigma_{max}^2\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right) + \frac{1}{2}\sigma_{max}^2(\beta\boldsymbol{V}) + \frac{1}{2}\sigma_{max}^2\left(\beta\boldsymbol{V}^{\mathrm{T}}\right)\right)\|\Delta\boldsymbol{W}\|_F^2
\end{aligned} \tag{25}$$

Where $\Delta\boldsymbol{W} = \boldsymbol{W}_1 - \boldsymbol{W}_2$, $\sigma_{max}(\cdot)$ is the largest singular value of the matrix, then:

$$\|\nabla f(\boldsymbol{W}_1) - \nabla f(\boldsymbol{W}_2)\|_F^2 \le \left(2\sigma_{max}^2\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right) + \frac{1}{2}\sigma_{max}^2(\beta\boldsymbol{V}) + \frac{1}{2}\sigma_{max}^2\left(\beta\boldsymbol{V}^{\mathrm{T}}\right)\right)\|\Delta\boldsymbol{W}\|_F^2 \tag{26}$$

Therefore, the Lipschitz constant is

$$L_f = \sqrt{2\sigma_{max}^2\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right) + \frac{1}{2}\sigma_{max}^2(\beta\boldsymbol{V}) + \frac{1}{2}\sigma_{max}^2\left(\beta\boldsymbol{V}^{\mathrm{T}}\right)} \tag{27}$$

3 Pseudocode and complexity analysis

3.1 Pseudocode

The overall DC method is summarized in Algorithm 1.

Algorithm 1: The DC Method

Input: label matrix $\boldsymbol{Y}$, label count n

Output: label adjacency matrix $\boldsymbol{C}$

1) for $i = 1, 2, 3, \cdots, n$ do
2) for $j = 1, 2, 3, \cdots, n$ do
3) construct $P(y_i)$ *and* $P(y_j|y_i)$, calculate $D_{y_i \to y_j} = D(P(y_i), P(y_j|y_i))$
4) construct $P(y_j)$ *and* $P(y_i|y_j)$, calculate $D_{y_j \to y_i} = D(P(y_j), P(y_i|y_j))$
5) end for
6) end for
7) if $D_{y_j \to y_i} > D_{y_i \to y_j}$, then $y_i \to y_j$ is the inferred direction, $\boldsymbol{C}_{ij} = 1$
8) if $D_{y_i \to y_j} > D_{y_j \to y_i}$, then $y_j \to y_i$ is the inferred direction, $\boldsymbol{C}_{ji} = 1$
9) otherwise, $\boldsymbol{C}_{ij} = 0$, no decision is made.

The accelerated proximal gradient of ACML is summarized in Algorithm 2.

Algorithm 2: The Accelerated Proximal Gradient Method

Input: training set $\boldsymbol{X}$, ***label matrix*** $\boldsymbol{Y}$, parameters α, β, τ

Output: coefficient matrix $\boldsymbol{W}$

1) initialization: $b_0 = b_1 = 1, \boldsymbol{W}_0 = \boldsymbol{W}_1 = (\boldsymbol{X}^T\boldsymbol{X} + \tau I)^{-1}\boldsymbol{X}^T\boldsymbol{Y}$
2) **while** not converged **do**
3) $\boldsymbol{W}^{(t)} = \boldsymbol{W}_t + \frac{b_{t-1}-1}{b_t}(\boldsymbol{W}_t - \boldsymbol{W}_{t-1})$
4) $G^{(t)} = \boldsymbol{W}^t - \frac{1}{L_g}\nabla f\left(\boldsymbol{W}^{(t)}\right)$
5) $\boldsymbol{W}_{t+1} = S_{\frac{\beta}{L_g}}\left(G^{(t)}\right)$
6) $b_{t+1} = \frac{1+\sqrt{4b_t^2+1}}{2}, t = t+1$
7) $t = t+1$
8) **end whiles**

Use ACML as a multi-label classifier is summarized in Algorithm 3.

Algorithm 3: ACML

Input: testing data $\boldsymbol{X}$, coefficient matrix $\boldsymbol{W}$, threshold t

Output: predict label matrix $\boldsymbol{Y}^*$, score matrix $\boldsymbol{S}$

1) $\boldsymbol{S} = \boldsymbol{X}\boldsymbol{W}$
2) $\boldsymbol{Y}^* = \mathrm{sign}(\boldsymbol{S} - t)$

3.2 Complexity analysis

In algorithm 1, The label adjacency matrix $\boldsymbol{C}$ is construct with the DC method. Only the non-diagonal elements in the upper or lower triangular matrix need to be calculate. Therefore, the complexity of step 3 and step 4 is $O(l^2)$. Step 5 have a complexity of $O(l^2/2)$.

In algorithm 2, The most time-consuming steps are step 4 and step 5. In step 4, the complexity of solving $\boldsymbol{W}$ is $O(nd^2l + nl^2 + dl^2)$. In step 5, the complexity of calculating Lipschitz constant is $O(nd^2 + d^2l)$.

In summary, the time complexity of the ACML algorithm is $O(d^2(nl + n + l) + l^2(n + d + 3/2))$. The most time-consuming steps of the ACML algorithm are, the DC method to calculate the label adjacency matrix $\boldsymbol{C}$, the accelerated gradient descent updating the coefficient matrix $\boldsymbol{W}$ and calculate the Lipschitz constant. In comparison experiments, the performance of LSML algorithm [18] and FF-IMLL algorithm [7] are close to the proposed ACML algorithm. Among them, the original paper of FF-IMLL algorithm does not give a specific algorithm complexity analysis. The complexity of LSML algorithm is $O((n + l)d^2 + (n + d)l^2 + ndl + d^3 + l^3)$. The complexity of the proposed ACML algorithm is slightly lower than LSML algorithm. The experimental results show that the ACML algorithm in this paper is superior to LSML in most evaluation metrics, and the LSML algorithm is only superior in some metrics of some datasets.

4 Experiment

4.1 Dataset

In order to validate the effectiveness of the proposed algorithm, 13 public multi-label benchmark datasets are selected from Mulan. The number of instances is from 978 to 5000, the number of labels is from 22 to 53, and the number of features is from 438 to 1449. Therefore, the selected dataset has a certain representativeness, and the specific dataset description is shown in Table 1.

4.2 Metric

This paper selects 5 commonly used multi-label algorithm evaluation metrics [19–21] to evaluate the performance of the ACML algorithm and the comparison algorithms. $D = \{(X_{it}, Y_{il} | 1 \le t \le d, 1 \le i \le n, 1 \le l \le L)\}$ is a multi-label dataset, $h(\cdot)$ is a multi-label classifier, $f(\cdot, \cdot)$ is a prediction function, $rank_f$ is a ranking function, the definitions of the selected 5 evaluation metrics are given below:

Average Precision: The calculation of average precision (AP) is shown in Eq. (28). The average precision measures the classification accuracy. Average precision combines the precision and recall, where the precision reflects the accuracy prediction result, while recall reflects the accuracy of prediction for positive samples. The larger the $AP_D(f)$ is, the better the classifier $f(\cdot, \cdot)$ performance, and the best performance when $AP_D(f) = 1$.

$$AP_D(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|}\cdot\sum_{y\in Y_i}\frac{\left|\left\{y' | rank_f(\boldsymbol{x}_i, y') \le rank_f(\boldsymbol{x}_i, y), y' \in Y_i\right\}\right|}{rank_f(\boldsymbol{x}_i, y)} \tag{28}$$

Table 1 Description of datasets

Dataset	Instance	Feature	Label	Domain
Arts[1]	5000	462	26	Text
Business[1]	5000	438	30	Text
Computers[1]	5000	681	33	Text
Education[1]	5000	550	33	Text
Enron[1]	1702	1001	53	Text
Entertainment[1]	5000	640	21	Text
Health[1]	5000	612	32	Text
Medical[1]	978	1449	45	Text
Recreation[1]	5000	606	22	Text
Reference [1]	5000	793	33	Text
Science[1]	5000	743	40	Text
Social[1]	5000	1047	39	Text
Society[1]	5000	636	27	Text

[1] Mulan http://mulan.sourceforge.net/datasets-mlc.html

Hamming Loss: The calculation of Hamming loss (HL) is shown in Eq. (29). Hamming loss evaluates the difference between the ground truth and the prediction. That is, the number of misclassifications of each label in the test set. Where, Δ is the symmetric difference between two sets. The smaller the $HL_D(h)$ is, the better the classifier $f(\cdot, \cdot)$ performance, and the best performance when $HL_D(h) = 0$.

$$HL_D(h) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{|Y|}|h(\boldsymbol{x}_i)\Delta Y_i|\right) \tag{29}$$

Ranking Loss: The calculation of ranking loss (RL) is shown in Eq. (30). The ranking loss compares the positive label set and the negative label set in pairwise, and counts that the ranking of the positive label is lower than the ranking of the negative label. That is, the ranking loss considers the situation when irrelevant labels in the prediction result are ranked higher than related labels. The smaller the $RL_D(f)$ is, the better the classifier $f(\cdot, \cdot)$ performance, and the best performance when $RL_D(f) = 0$.

$$L_D(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|\left|\overline{Y_i}\right|}\cdot\left|\left\{(y_1, y_2) | f(\boldsymbol{x}_i, y_1) \le f(\boldsymbol{x}_i, y_2), (y_1, y_2) \in Y_i \times \overline{Y_i}\right\}\right| \tag{30}$$

One-Error: The calculation of one-error (OE) is shown in Eq. (31). One-error reflects the error rate of prediction with highest ranked (Top-Ranked Label). The smaller the $OE_D(f)$ is, the better the classifier $f(\cdot, \cdot)$ performance, and the best performance when $OE_D(f) = 0$.

$$OE_D(f) = \frac{1}{n}\sum_{i=1}^{n}\left[\left[\arg\max_{y\in Y} f(\boldsymbol{x}_i, y)\right] \notin Y_i\right] \tag{31}$$

Coverage: The calculation of coverage (CV) is shown in Eq. (32). The coverage indicates how many top-scored prediction labels must include without left out any ground truth label. The smaller the $CV_D(f)$ is, the better the classifier $f(\cdot, \cdot)$ performance, and the best performance when $CV_D(f) = 0$.

$$CV_D(f) = \frac{1}{n}\sum_{i=1}^{n}\max_{y\in Y_i} rank_f(\boldsymbol{x}_i, y) - 1 \tag{32}$$

4.3 Parameter

In the comparison experiment, we selected a total of 5 multi-label classification algorithms for comparison. The 5 comparison algorithms selected in this paper all consider the symmetric label correlation from different

Table 2 AP results of each algorithm on 13 datasets

Dataset	ACML	LLSF	LSF-CI	LSML	GroPLE	FF-IMLL
Arts	**0.6241 ± 0.0141**	0.5852 ± 0.0147	0.5451 ± 0.0100	0.5932 ± 0.0069	0.6195 ± 0.0041	0.5291 ± 0.0153
Business	**0.8809 ± 0.0072**	0.8484 ± 0.0090	0.7825 ± 0.0116	0.8798 ± 0.0085	0.8684 ± 0.0074	0.8793 ± 0.0149
Computers	**0.7093 ± 0.0163**	0.6575 ± 0.0064	0.5839 ± 0.0059	0.6915 ± 0.0059	0.3825 ± 0.0243	0.6416 ± 0.0092
Education	**0.6337 ± 0.0153**	0.5805 ± 0.0069	0.5290 ± 0.0166	0.6033 ± 0.0082	0.5841 ± 0.0116	0.5543 ± 0.0074
Enron	**0.7027 ± 0.0099**	0.6609 ± 0.0085	0.6539 ± 0.0060	0.6928 ± 0.0088	0.5099 ± 0.0089	0.6531 ± 0.0113
Entertainment	**0.6925 ± 0.0067**	0.6669 ± 0.0071	0.6351 ± 0.0076	0.6731 ± 0.0089	0.6501 ± 0.0115	0.5815 ± 0.0137
Health	**0.7821 ± 0.0062**	0.7571 ± 0.0059	0.6980 ± 0.0094	0.7646 ± 0.0098	0.7216 ± 0.0096	0.7138 ± 0.0055
Medical	**0.9012 ± 0.0224**	0.8668 ± 0.0190	0.8696 ± 0.0107	0.9008 ± 0.0129	0.8575 ± 0.0061	0.8273 ± 0.0150
Recreation	**0.6391 ± 0.0043**	0.5985 ± 0.0148	0.5692 ± 0.0097	0.6185 ± 0.0097	0.6095 ± 0.0128	0.4924 ± 0.0047
Reference	**0.7135 ± 0.0033**	0.6634 ± 0.0129	0.5929 ± 0.0173	0.7052 ± 0.0072	0.6477 ± 0.0076	0.6290 ± 0.0141
Science	**0.6077 ± 0.0081**	0.5521 ± 0.0105	0.5166 ± 0.0116	0.5890 ± 0.0158	0.5623 ± 0.0066	0.4570 ± 0.0122
Social	**0.7769 ± 0.0053**	0.7392 ± 0.0051	0.6993 ± 0.0199	0.7587 ± 0.0082	0.7076 ± 0.0072	0.7047 ± 0.0089
Society	**0.6360 ± 0.0076**	0.5930 ± 0.0071	0.5158 ± 0.0108	0.6128 ± 0.0118	0.5625 ± 0.0048	0.5920 ± 0.0083

perspective. Among them, LLSF algorithm [22] uses the cosine similarity to obtain label correlation to construct a sparse coefficient matrix to learn label-specific features. LSF-CI algorithm considers the similarity of instance features based on the k-nearest neighbor and LLSF algorithm, which further improves the algorithm efficiency. The LSML algorithm classifies multi-label data by learning high-order label correlation matrix. GroPLE algorithm [23] embeds features and labels into low-dimensional space to achieve multi-label classification. FF-MLLA algorithm measures the similarity between instances by calculating the distance and finds the nearest neighbors of each label. On this basis, it combines with the Firefly algorithm to transform the label counting matrix and perform multi-label learning.

The parameter settings of each algorithm in the comparison experiment are given as below:

The parameter settings of the ACML algorithm are $\alpha\epsilon[2^{-10}, 2^{10}]$, $\beta\epsilon[2^{-10}, 2^{10}]$.
The parameter settings of the LLSF algorithm are $\alpha = [2^{-10}, 2^{10}]$, $\beta = [2^{-10}, 2^{10}]$, $\gamma = \{0.1,10\}$.
The parameter settings of the LSF-CI algorithm are $\alpha = [2^{-10}, 2^{10}]$, $\beta = [2^{-12}, 2^{12}]$, $\gamma = \{0.1,10\}$.
The parameter settings of the LSML algorithm are λ_1, λ_2, λ_3, $\lambda_4 = [10^{-5}, 10^{3}]$.
The parameter settings of the GroPLE algorithm are $\alpha \in [10^{-4}, 10^{4}]$, $\beta \in [10^{-4}, 10^{4}]$, $\gamma \in [10^{-2}, 10^{2}]$, $\lambda_1 \in [10^{-3}, 10^{2}]$. $\lambda_2 \in [10^{-3}, 10^{2}]$.

Table 3 HL results of each algorithm on 13 datasets

Dataset	ACML	LLSF	LSF-CI	LSML	GroPLE	FF-IMLL
Arts	**0.0536 ± 0.0007**	0.0566 ± 0.0009	0.0561 ± 0.0013	0.0582 ± 0.0011	0.0560 ± 0.004	0.0590 ± 0.0015
Business	0.0266 ± 0.0004	0.0295 ± 0.0009	0.0398 ± 0.0008	0.0287 ± 0.0010	0.0278 ± 0.0016	**0.0264 ± 0.0024**
Computers	**0.0339 ± 0.0009**	0.0389 ± 0.0010	0.0415 ± 0.0018	0.0391 ± 0.0008	0.5097 ± 0.0314	0.0382 ± 0.0009
Education	**0.0371 ± 0.0008**	0.0414 ± 0.0007	0.0418 ± 0.0012	0.0411 ± 0.0003	0.0720 ± 0.0009	0.0406 ± 0.0007
Enron	**0.0455 ± 0.0006**	0.0553 ± 0.0020	0.0664 ± 0.0066	0.0505 ± 0.0014	0.1462 ± 0.0246	0.0504 ± 0.0023
Entertainment	**0.0508 ± 0.0014**	0.0550 ± 0.0015	0.0550 ± 0.0014	0.0570 ± 0.0006	0.1089 ± 0.0021	0.0592 ± 0.0018
Health	**0.0331 ± 0.0008**	0.0356 ± 0.0010	0.0389 ± 0.0009	0.0413 ± 0.0006	0.0695 ± 0.0024	0.0400 ± 0.0083
Medical	**0.0114 ± 0.0017**	0.0166 ± 0.0070	0.0219 ± 0.0059	0.0156 ± 0.0011	0.0205 ± 0.0012	0.5316 ± 0.0206
Recreation	**0.0535 ± 0.0011**	0.0571 ± 0.0013	0.0565 ± 0.0003	0.0578 ± 0.0008	0.1166 ± 0.0030	0.0600 ± 0.0025
Reference	**0.0257 ± 0.0008**	0.0280 ± 0.0006	0.0298 ± 0.0014	0.0294 ± 0.0010	0.0722 ± 0.0035	0.0292 ± 0.0016
Science	**0.0311 ± 0.0006**	0.0348 ± 0.0007	0.0348 ± 0.0007	0.0333 ± 0.0007	0.0797 ± 0.0013	0.0350 ± 0.0011
Social	**0.0203 ± 0.0003**	0.0232 ± 0.0008	0.0241 ± 0.0009	0.0244 ± 0.0006	0.0596 ± 0.0007	0.0244 ± 0.0010
Society	**0.0515 ± 0.0010**	0.0578 ± 0.0011	0.0585 ± 0.0013	0.0560 ± 0.0010	0.1183 ± 0.0026	0.0554 ± 0.0023

Table 4 RL results of each algorithm on 13 datasets

Dataset	ACML	LLSF	LSF-CI	LSML	GroPLE	FF-IMLL
Arts	**0.1405 ± 0.0074**	0.1841 ± 0.0106	0.2621 ± 0.0106	0.1770 ± 0.0058	0.1273 ± 0.0028	0.1558 ± 0.0049
Business	**0.0443 ± 0.0048**	0.0635 ± 0.0041	0.1036 ± 0.0071	0.0485 ± 0.0048	0.0526 ± 0.0051	0.0451 ± 0.0063
Computers	0.0980 ± 0.0086	0.1229 ± 0.0059	0.2299 ± 0.0126	0.1230 ± 0.0059	0.3749 ± 0.0256	**0.0974 ± 0.0036**
Education	0.1089 ± 0.0057	0.1642 ± 0.0065	0.2486 ± 0.0081	0.1526 ± 0.0068	0.1981 ± 0.0114	**0.1020 ± 0.0024**
Enron	**0.0895 ± 0.0039**	0.1258 ± 0.0059	0.1109 ± 0.0023	0.0951 ± 0.0043	0.2368 ± 0.0137	0.1024 ± 0.0059
Entertainment	**0.1163 ± 0.0035**	0.1460 ± 0.0110	0.2215 ± 0.0096	0.1422 ± 0.0040	0.1804 ± 0.0104	0.1272 ± 0.0066
Health	0.0677 ± 0.0041	0.0905 ± 0.0034	0.1651 ± 0.0060	0.0876 ± 0.0044	0.1403 ± 0.0053	**0.0659 ± 0.0021**
Medical	**0.0150 ± 0.0032**	0.0442 ± 0.0116	0.0479 ± 0.0033	0.0185 ± 0.0064	0.0414 ± 0.0091	0.0321 ± 0.0029
Recreation	**0.1485 ± 0.0037**	0.1868 ± 0.0045	0.2446 ± 0.0075	0.1741 ± 0.0107	0.1966 ± 0.0084	0.1889 ± 0.0055
Reference	**0.0930 ± 0.0047**	0.1398 ± 0.0087	0.2426 ± 0.0149	0.1070 ± 0.0060	0.1700 ± 0.0052	0.0933 ± 0.0032
Science	**0.1289 ± 0.0046**	0.1774 ± 0.0083	0.2473 ± 0.0056	0.1530 ± 0.0106	0.1916 ± 0.0059	0.1574 ± 0.0065
Social	**0.0715 ± 0.0049**	0.0924 ± 0.0072	0.1830 ± 0.0158	0.0897 ± 0.0044	0.1511 ± 0.0027	0.0749 ± 0.0042
Society	0.1517 ± 0.0061	0.1857 ± 0.0042	0.2820 ± 0.0071	0.1879 ± 0.0054	0.2337 ± 0.0041	**0.1495 ± 0.0023**

The parameter settings of the FF-MLLA algorithm are clusters size = [5, 15], kernel size = [10, 100.]

5 Analysis and visualization

5.1 Experimental results

This paper uses 5-fold cross-validation to evaluate the performance of each comparison algorithm. Cross-validation refers to randomly assigning a dataset to n subsets with equal size. These n subsets are used as the test set in turn, and the remaining subsets are used for training. 5-fold cross-validation refers to the situation where n=5. Table 2-6 shows the specific performance of the algorithm proposed in this paper and other 5 comparison algorithms on the selected 5 evaluation metrics. In Tables 2, 3, 4, 5, 6, and 7, the best results are expressed in bold text.

It can be seen from Table 2 that the proposed algorithm is significantly performed better than other comparison algorithms in average accuracy, the performance of the ACML algorithm on all 13 benchmark datasets is better than other comparison algorithms. At the same time, the variance of the ACML is smaller, indicating that the performance of the ACML is more stable. As shown in Table 3, the ACML algorithm is generally better than other comparison algorithms in terms of Hamming loss, and only slightly falls behind the FF-IMLL algorithm on the Business dataset. Similar to the average accuracy, the ACML's overall variance of Hamming loss is smaller, the stability of the ACML algorithm is better. The

Table 5 OE results of each algorithm on 13 datasets

Dataset	ACML	LLSF	LSF-CI	LSML	GroPLE	FF-IMLL
Arts	**0.4524 ± 0.0179**	0.4900 ± 0.0181	0.5090 ± 0.0180	0.4762 ± 0.0088	0.4674 ± 0.0112	0.5988 ± 0.0232
Business	0.1158 ± 0.0074	0.1452 ± 0.0112	0.2240 ± 0.0185	**0.1104 ± 0.0105**	0.1250 ± 0.0057	0.1164 ± 0.0171
Computers	**0.3466 ± 0.0171**	0.4080 ± 0.0094	0.4614 ± 0.0072	0.3608 ± 0.0060	0.6941 ± 0.0253	0.4332 ± 0.0144
Education	**0.4606 ± 0.0203**	0.5090 ± 0.0070	0.5290 ± 0.0166	0.4826 ± 0.0178	0.4902 ± 0.0150	0.5800 ± 0.0133
Enron	**0.2279 ± 0.0216**	0.2650 ± 0.0147	0.2644 ± 0.0114	0.2444 ± 0.0156	0.4301 ± 0.0124	0.2685 ± 0.0248
Entertainment	**0.3912 ± 0.0106**	0.4092 ± 0.0070	0.4166 ± 0.0063	0.4072 ± 0.0113	0.4170 ± 0.0141	0.5604 ± 0.0172
Health	**0.2510 ± 0.0068**	0.2764 ± 0.0141	0.3172 ± 0.0162	0.2644 ± 0.0109	0.2936 ± 0.0089	0.3642 ± 0.0101
Medical	0.1381 ± 0.0371	0.1697 ± 0.0208	0.1605 ± 0.0199	**0.1350 ± 0.0132**	0.1928 ± 0.0138	0.2301 ± 0.0216
Recreation	**0.4444 ± 0.0092**	0.4890 ± 0.0248	0.5056 ± 0.0113	0.4614 ± 0.0102	0.4616 ± 0.0120	0.6496 ± 0.0031
Reference	**0.3642 ± 0.0090**	0.4020 ± 0.0144	0.4692 ± 0.0178	0.3666 ± 0.0088	0.4202 ± 0.0094	0.4686 ± 0.0160
Science	**0.4772 ± 0.0108**	0.5274 ± 0.0075	0.5512 ± 0.0191	0.4884 ± 0.0207	0.5056 ± 0.0130	0.6652 ± 0.0130
Social	**0.2710 ± 0.0040**	0.3108 ± 0.0111	0.3204 ± 0.0207	0.2858 ± 0.0112	0.3266 ± 0.0110	0.3840 ± 0.0078
Society	**0.3900 ± 0.0106**	0.4462 ± 0.0099	0.5192 ± 0.0192	0.4028 ± 0.0183	0.4656 ± 0.0077	0.4570 ± 0.0073

Table 6 CV results of each algorithm on 13 datasets

Dataset	ACML	LLSF	LSF-CI	LSML	GroPLE	FF-IMLL
Arts	**0.2141 ± 0.0080**	0.2650 ± 0.0117	0.3448 ± 0.0107	0.2567 ± 0.0060	0.1958 ± 0.0056	0.2192 ± 0.0084
Business	0.0895 ± 0.0078	0.1096 ± 0.0064	0.1559 ± 0.0092	0.0967 ± 0.0097	0.1020 ± 0.0079	**0.0841 ± 0.0094**
Computers	0.1406 ± 0.0106	0.1720 ± 0.0088	0.2888 ± 0.0170	0.1725 ± 0.0051	0.4326 ± 0.0224	**0.1401 ± 0.0030**
Education	0.1592 ± 0.0086	0.2215 ± 0.0068	0.3133 ± 0.0115	0.2123 ± 0.0077	0.2665 ± 0.0154	**0.1356 ± 0.0042**
Enron	**0.2547 ± 0.0072**	0.3154 ± 0.0156	0.2779 ± 0.0045	0.2695 ± 0.0100	0.5102 ± 0.7442	0.2810 ± 0.0130
Entertainment	**0.1605 ± 0.0066**	0.1918 ± 0.0114	0.2717 ± 0.0109	0.1897 ± 0.0044	0.2323 ± 0.0098	0.1649 ± 0.0091
Health	0.1286 ± 0.0072	0.1562 ± 0.0035	0.2456 ± 0.0067	0.1591 ± 0.0060	0.2250 ± 0.0078	**0.1131 ± 0.0045**
Medical	**0.0257 ± 0.0033**	0.0580 ± 0.0130	0.0561 ± 0.0074	0.0294 ± 0.0091	0.0538 ± 0.0103	0.0451 ± 0.0063
Recreation	**0.1992 ± 0.0059**	0.2392 ± 0.0044	0.2968 ± 0.0088	0.2277 ± 0.0123	0.2519 ± 0.0082	0.2293 ± 0.0073
Reference	0.1194 ± 0.0079	0.1705 ± 0.0104	0.2745 ± 0.0171	0.1354 ± 0.0072	0.2045 ± 0.0043	**0.1105 ± 0.0042**
Science	**0.1778 ± 0.0058**	0.2296 ± 0.0087	0.3027 ± 0.0065	0.2048 ± 0.0143	0.2477 ± 0.0073	0.1991 ± 0.0089
Social	0.1035 ± 0.0064	0.1298 ± 0.0081	0.2210 ± 0.0170	0.1268 ± 0.0053	0.1975 ± 0.0024	**0.1015 ± 0.0063**
Society	0.2387 ± 0.0115	0.2755 ± 0.0061	0.3820 ± 0.0090	0.2847 ± 0.0064	0.3340 ± 0.0056	**0.2223 ± 0.0095**

comparison of the ranking loss is shown in Table 4. The ACML algorithm is overall dominant, and only slightly behind the FF-IMLL algorithm on some datasets. FF-IMLL further calculates the difference between the different labels in the neighbor points while considering the neighbor points. Which makes the FF-IMLL algorithm have a better prediction performance on the labels with higher correlation. As shown in Table 5, the ACML algorithm is partially better on the one-error metric, and only has a certain difference with the LSML algorithm on the Business and Medical datasets. The result of coverage is shown in Table 6. The proposed algorithm is better in half datasets, and the FF-IMLL algorithm is superior to the algorithm in half of the datasets. This is because the FF-IMLL algorithm considers the importance of the nearest neighbor labels, making the FF-IMLL algorithm has a higher numerical output when predicting related labels, the ranking of the positive label is higher, so the coverage is smaller.

Based on 5 evaluation metrics, it can be found that the proposed algorithm is superior to other comparison algorithms in most cases, indicating that the ACML algorithm has performance advantages and high stability. At the same time, in order to verify that the introduction of the asymmetric label correlation in the model improves the performance of the algorithm. We compare the ACML using the asymmetric label correlation matrix with the ACML-Naive using the label correlation matrix. Some results are as shown in Table 7, the performance of the ACML algorithm using the asymmetric label correlation matrix is better than the ACML-Naïve

Table 7 Part results of ACML and ACML-Naive on 13 datasets

Dataset	ACML	ACML-Naive	ACML	ACML-Naive	ACML	ACML-Naive
Arts	**0.6241±0.0141**	0.6208±0.0078	**0.0536±0.0007**	0.0537±0.0019	**0.2141±0.0080**	0.2146±0.0066
Business	**0.8809±0.0072**	0.8781±0.0043	**0.0266±0.0004**	0.0268±0.0008	**0.0895±0.0078**	0.0892±0.0034
Computers	**0.7093±0.0163**	0.7050±0.0112	**0.0339±0.0009**	0.0339±0.0010	**0.1406±0.0106**	0.1413±0.0052
Education	**0.6337±0.0153**	0.6308±0.0040	**0.0371±0.0008**	0.0372±0.0006	**0.1592±0.0086**	0.1633±0.0056
Enron	**0.7027±0.0099**	0.6973±0.0067	**0.0455±0.0006**	0.0461±0.0018	**0.2547±0.0072**	0.2588±0.0119
Entertainment	**0.6925±0.0067**	0.6886±0.0042	**0.0508±0.0014**	0.0511±0.0009	**0.1605±0.0066**	0.1638±0.0038
Health	**0.7821±0.0062**	0.7796±0.0048	**0.0331±0.0008**	0.0334±0.0004	**0.1286±0.0072**	0.1298±0.0062
Medical	**0.9012±0.0224**	0.8971±0.0212	**0.0114±0.0017**	0.0113±0.0014	**0.0257±0.0033**	0.0258±0.0105
Recreation	**0.6391±0.0043**	0.6325±0.0105	**0.0535±0.0011**	**0.0535±0.0011**	**0.1992±0.0059**	0.2022±0.0062
Reference	**0.7135±0.0033**	0.7092±0.0088	0.0257±0.0008	**0.0255±0.0009**	**0.1194±0.0079**	0.1209±0.0091
Science	**0.6077±0.0081**	0.6052±0.0034	0.0311±0.0006	**0.0310±0.0005**	0.1778±0.0058	**0.1771±0.0074**
Social	**0.7769±0.0053**	0.7747±0.0031	**0.0203±0.0003**	0.0204±0.0004	**0.1035±0.0064**	0.1043±0.0031
Society	**0.6360±0.0076**	0.6341±0.0075	**0.0515±0.0010**	0.0516±0.0018	**0.2387±0.0115**	0.2403±0.0077
Metric	AP		HL		CV	

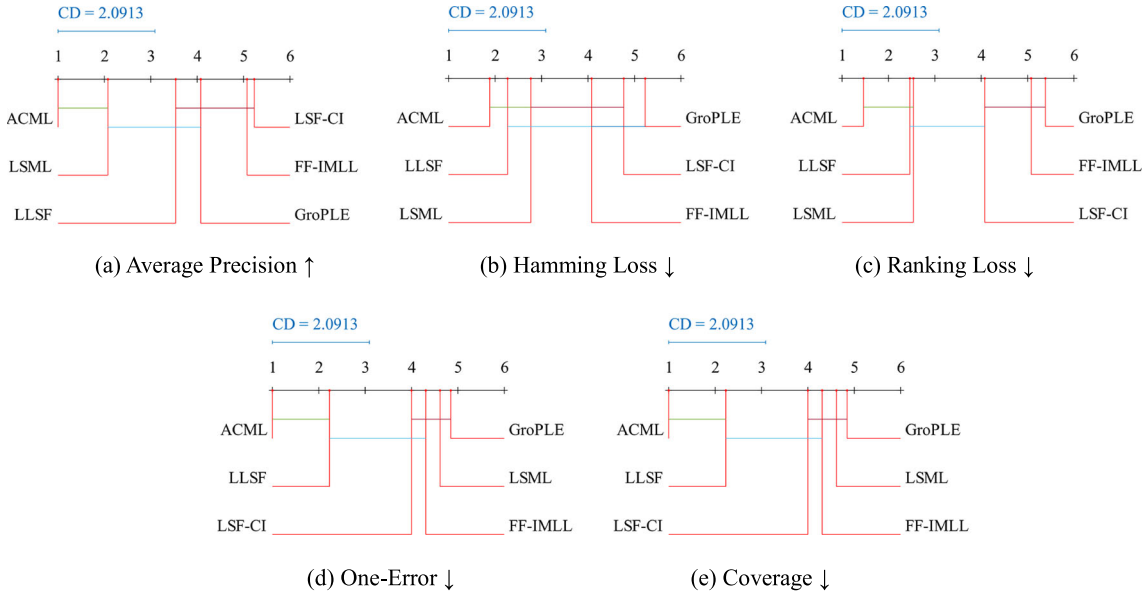


Fig. 2 Nemenyi test ($\alpha = 0.05$). **a** Average Precision ↑, **b** Hamming Loss ↓, **c** Ranking Loss ↓, **d** One-Error ↓, **e** Coverage ↓

algorithm using the label correlation matrix. It furtherly illustrates the effectiveness and rationality of introducing asymmetric label correlation in the multi-label algorithm.

5.2 Hypothesis testing

In this section, we will further verify the effectiveness of this algorithm through hypothesis testing. This paper uses the Nemenyi test to compare the algorithms. The equation of the Nemenyi test is as follows:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{7N}} \tag{33}$$

The significance level $\alpha = 0.05$, the algorithm involved in the comparison $k = 6$, the dataset $N = 13$, and the critical difference $q_\alpha = 2.949$. It can be seen from Fig. 2 that compared with the comparison algorithm, the ACML is algorithm dominant in most indicators. Among them, the average accuracy of the ACML is significantly different from most of other comparison algorithms. The ACML algorithm is also significantly better than the FF-IMLL, GroPLE, LSF-CI in Hamming loss. The ranking loss of the ACML is significantly different from FF-IMLL, LSF-CI, and GroPLE. In one-error metric, the ACML has significant differences with most of other comparison algorithms. In terms of coverage, ACML is not significantly different from LSML and LLSF, but there are significant differences from FF-IMLL, LSF-CI, and GroPLE. The results of Nemenyi test basically consistent with the results of the experimental analysis. The results of Nemenyi test further verify the performance of the ACML algorithm, and show that the introduction of asymmetric label correlation in multi-label learning is reasonable and effective.

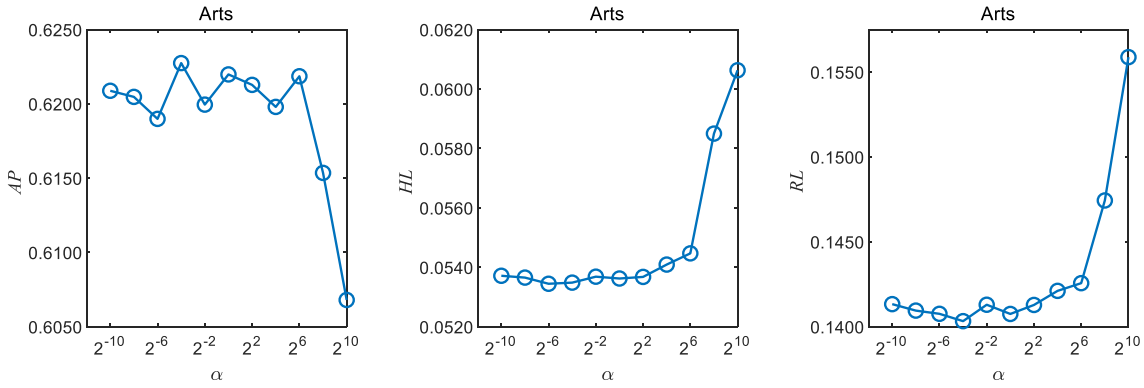
5.3 Parameter sensitivity

The proposed algorithm has two main parameters, parameter α is the sparse parameter, which controls the sparsity of the learned model. Parameter β controls the effect of asymmetric label correlation of the model. We select Arts and Medical datasets to perform parameter sensitivity analysis. Parameter sensitivity analysis is performed under three selected metrics, including AP, HL, and RL. We fix one parameter and vary another parameter to select the best combination of parameters.

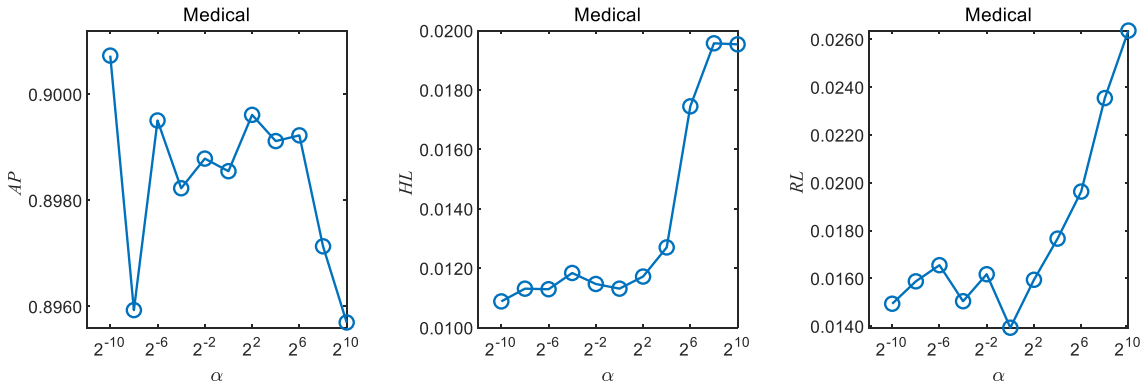
For parameter α, we fix β=2^{-10} and adjust the parameter α. The specific analysis results are shown in Fig. 3(a)-(b). It can be found that in both datasets, when α values around α=2^{-2}, the comprehensive effect of the three evaluation metrics is the best. From the experimental results, it can be found that only when the value of the parameter α is moderate, the algorithm has more stable performance. This is because when the sparsity is too large, the model will ignore too much detailed information. When the sparsity is insufficient, it cannot extract effective feature information.

For parameter β, we fix α=2^{-2} and perform sensitivity analysis on the parameter β. Figure 3(c)-(d) shows the sensitivity analysis of parameter β. On the Arts dataset, a bigger β leads to better performance. While the influence of β on the

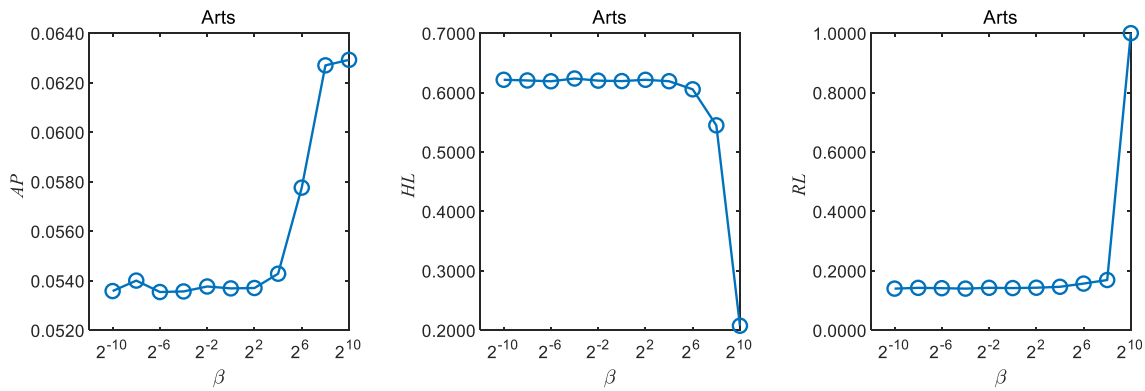
Fig. 3 Parameter sensitivity analysis on Arts and Medical. **a** Sensitivity analysis of parameter α in the Arts dataset, **b** Sensitivity analysis of parameter α in the Medical dataset, **c** Sensitivity analysis of parameter β in the Arts dataset, **d** Sensitivity analysis of parameter β in the Medical dataset



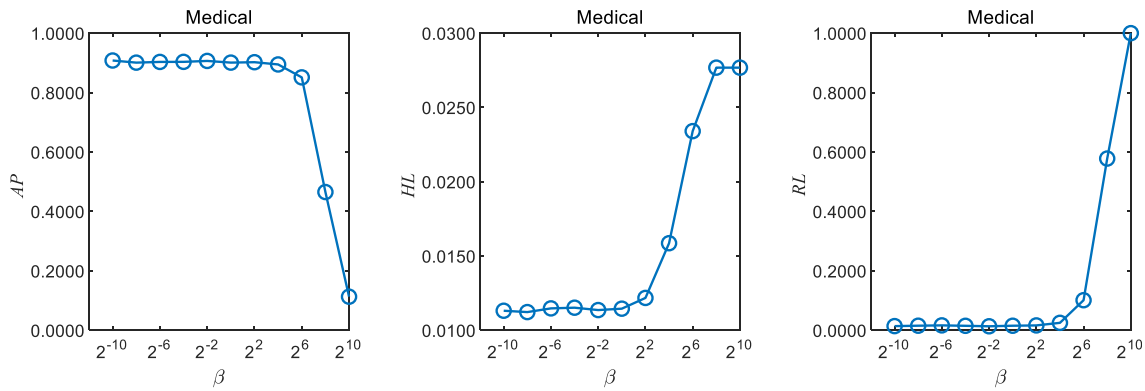
(a) Sensitivity analysis of parameter α in the Arts dataset



(b) Sensitivity analysis of parameter α in the Medical dataset



(c) Sensitivity analysis of parameter β in the Arts dataset



(d) Sensitivity analysis of parameter β in the Medical dataset

Medical dataset is almost opposite to that on the Arts dataset. One possible explanation is that in Arts dataset, asymmetric label correlation strongly exists. Therefore, when β increases, the model performance improves accordingly. In contrast, there is no significant asymmetric label correlation between the labels of the Medical dataset, so a larger parameter β will reduce the performance of the model.

6 Conclusion

Currently, most existing multi-label learning algorithms employ label correlation as an important method for learning information from the label space. By considering label correlation, algorithms can mine correlations between labels and improve the performance of the classifier. However, simply considering the correlation between labels and ignoring the asymmetry in the correlation relationship sometimes introduces redundant information into the model. Such redundant information increases the complexity of the model and reduces the performance of the classifier. In order to effectively consider the asymmetry between labels, this paper proposes to use DC method to infer the asymmetry between labels. By introducing the label asymmetric label correlation matrix into the model, the performance of multi-label learning is improved. Compared with other state-of-art multi-label learning algorithms with label correlation, the experimental results show the effectiveness and rationality of the proposed algorithm. The proposed algorithm considers the asymmetric correlation between pairwise labels from a global perspective. However, it fails to consider the complex relationship among labels from both global and local perspectives. In addition, manually labeling often resulting in missing labels, which also affects the correctness of asymmetric label correlation. Therefore, how to use the graph method to consider the asymmetric label correlation in incomplete labeled data is the next important research direction.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China under Grant 61702012 and Key Laboratory of Data Science and Intelligence Application, Fujian Province University (NO. D202005) and Key Laboratory of Intelligent Computing & Signal Processing, Ministry of Education (Anhui University) (No.2020A003).

References

1. Zhang ML, Zhou ZH (2013) A review on multi-label learning algorithms[J]. IEEE Trans Knowl Data Eng 26(8):1819–1837
2. Tsoumakas G, Katakis I (2007) Multi-label classification: an overview[J]. Int J Data Warehousing Min 3(3):1–13
3. Yu Y, Pedrycz W, Miao D (2014) Multi-label classification by exploiting label correlations[J]. Expert Syst Appl 41(6):2989–3004
4. Xue L, Jiang D, Wang R (2020) Learning semantic dependencies with channel correlation for multi-label classification[J]. Vis Comput 36(7):1325–1335
5. Han H, Huang M, Zhang Y, Yang X, Feng W (2019) Multi-label learning with label specific features using correlation information[J]. IEEE Access 7:11474–11484
6. Zhu Y, Kwok JT, Zhou ZH (2017) Multi-label learning with global and local label correlation[J]. IEEE Trans Knowl Data Eng 30(6): 1081–1094
7. Cheng YS, Qian K, Wang YB, Zhao DW (2019) Multi-label lazy learning approach based on firefly method[J]. J Comput Appl 039(005):1305–1311
8. Huang SJ, Yu Y, Zhou ZH (2012) Multi-label hypothesis reuse[A]. Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. 525–533
9. Zhang M L, Zhang K (2010) Multi-label learning by exploiting label dependency[A]. Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining[C]. Washington D. C., 999–1007
10. Székely GJ, Rizzo ML, Bakirov NK (2007) Measuring and testing dependence by correlation of distances. Ann Stat 35(6):2769–2794
11. Zhou Y, Zhang Y, Zhu L (2020) A projective approach to conditional independence test for dependent processes[J]. J Bus Econ Stat. https://doi.org/10.1080/07350015.2020.1826952
12. Srinivasan G, Hyman JD, Osthus DA et al (2018) Quantifying topological uncertainty in fractured systems using graph theory and machine learning[J]. Sci Rep 8(1):1–11
13. Li J, Li X, He D (2019) A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction[J]. IEEE Access 7:75464–75475
14. Mihaljević B, Bielza C, Larrañaga P (2018) Learning Bayesian network classifiers with completed partially directed acyclic graphs[C]//international conference on probabilistic graphical models. PMLR:272–283
15. Liu F, Chan L (2016) Causal inference on discrete data via estimating distance correlations[J]. Neural Comput 28(5):801–814
16. Beck A, Teboulle M (2009) Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems[J]. IEEE Trans Image Process 18(11):2419–2434
17. Ganesh A, Lin Z, Wright J, et al. (2009) Fast algorithms for recovering a corrupted low-rank matrix[C]//2009 3rd IEEE international workshop on computational advances in multi-sensor adaptive processing (CAMSAP). IEEE 213–216
18. Huang J, Qin F, Zheng X, Cheng Z, Yuan Z, Zhang W, Huang Q (2019) Improving multi-label classification with missing labels by learning label-specific features[J]. Inf Sci 492:124–146
19. Gibaja EL, Ventura S (2015) A tutorial on multi-label learning[J]. ACM Comput Surv (CSUR) 47(3):1–38
20. Sorower MS (2010) A literature survey on algorithms for multi-label learning[J]. Oregon State University. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.364.5612&rep=rep1&type=pdf
21. Tsoumakas G, Katakis I, Vlahavas I (2009) Mining multi-label data[M]. In: Data Mining and Knowledge Discovery Handbook. Springer, Boston. https://doi.org/10.1007/978-0-387-09823-4_34
22. Huang J, Li G, Huang Q, Wu X (2015) Learning Label Specific Features for Multi-Label Classification[C]// 2015 IEEE International Conference on Data Mining. IEEE 181–190
23. Kumar V, Pujari AK, Padmanabhan V, Kagita VR (2019) Group preserving label embedding for multi-label classification[J]. Pattern Recogn 90:23–34

Jiachao Bao Graduate student of computer and Information College, Anqing Normal University. His main research includes machine learning, data mining and statistics.

Yusheng Cheng Professor of computer and Information College, Anqing Normal University. He received his Ph.D. in the School of Computer and Information Science of Hefei University of Technology in 2007. His research interests concern the rough set theory and algorithm, semi supervised learning and data mining. He is the author of more than 50 papers in journals and conference proceedings such as Information Science, Knowledge-Based Systems, Neurocomputing, Applied Soft Computing, PAKDD and so on.

Yibin Wang Professor of computer and Information College, Anqing Normal University. The main research directions include multi label learning, machine learning and software security.