Applied Intelligence
https://doi.org/10.1007/s10489-021-02868-4

Bi-directional mapping for multi-label learning of label-specific features

Yi Tan[1] · Dong Sun[1] · Yu Shi[1] · Liuya Gao[1] · Qingwei Gao[1] · Yixiang Lu[1]

Accepted: 21 September 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract
In multi-label learning, scholars have proposed many multi-label learning algorithms that explore label-specific features in recent years. Previous studies tend to focus only on the forward projection of the instance feature space to the category label space to learn label-specific features for multi-label classification, and only simple correlations between labels are considered; however, the loss of discriminative information in the instance space and the essential connections between labels resulting from the reduction of feature dimensionality during forward projection are usually ignored. Based on the overall consideration, in this paper, we propose a bi-directional mapping for multi-label learning of label-specific features method(BDLS). Specifically, under a unified linear model for learning label-specific features for multi-label classification, we propose a novel reconstruction loss function to compensate for the loss of discriminative information generated during forward mapping. And we also propose an effective causal learning machine to explore the intrinsic causal relationships among labels for the purpose of mining the essential connections among labels. Experimental results and analysis on several multi-label datasets validate the effectiveness of our proposed method.

Keywords Multi-label learning · Label-specific features · Bi-directional mapping · Label causality

1 Introduction

Supervised learning is an important branch of machine learning, of which single-label learning (SLL) is a traditional supervised learning method. It assumes that each instance only matches a specific category label and predicts a unique category label for invisible instances [1]. In most cases, however, real-world objects are complex and multi-semantic. Multi-label learning is proposed to solve such problems by predicting a discrete set of labels for agnostic and complex objects [2]. Over recent years, Multi-label learning has been extensively used in several application areas, such as text classification [3, 4], video annotation [5], bioanalysis [6], and music sentiment classification [7, 8].

Many approaches on multi-label learning have been proposed by related scholars. One of them binary relevance (BR) [9] transforms the multi-label learning problem into multiple independent single-label learning problems for solving multi-label classification problems. ML-KNN [10] is developed from the traditional K-nearest neighbors (KNN) algorithm. Firstly, the K-nearest neighbors of the test samples are located in the training set. Then the number of neighbors belonging to the same labels is calculated based on the statistical information of the neighboring samples. Finally, the group of category labels of the test samples is predicted using the maximum a posteriori probability principle (MAP). Most of these algorithms are designed directly for multi-label classification, while the majority of multi-label datasets have the characteristics of high dimensionality and large capacity. For example, when text analysis of drug information is performed, the drug treatment symptoms in the text are used as the characteristics of the drug. That is, the symptoms of patient are used to predict the corresponding drug for treatment. When the wide range of drug properties leads to an overabundance of therapeutic symptoms, it can cause a sharp increase in the number of therapeutic symptoms features dimensions. To solve the high-dimensional problem

✉ Dong Sun
sundong@ahu.edu.cn

Yi Tan
ahutany@163.com

[1] School of Electrical Engineering and Automation, Anhui University, Hefei, China

Published online: 21 October 2021

of multi-label data features, the research on feature dimensionality reduction for multi-label learning is of great significance [11]. Feature dimensionality reduction can optimize the performance of multi-label learning algorithms and improve the efficiency of the algorithms [12].

A traditional multi-label information feature dimensionality reduction method is to learn the unique characteristics of each label. LIFT [13] is the earliest algorithm proposed for label-specific feature learning. It uses the k-means algorithm to perform cluster analysis for each category labels, i.e., to form clusters of positive and negative instances, and then learns the specific features of each label by calculating the distance between the centers of the clusters and the instances. However, it neglects the importance of label relevance for specific feature extraction, and there is no strong interpretability. SFUS [14] is a multi-label learning method based on subspace sparse feature selection, which learns the most relevant features in the data by sparsity and mines the shared subspace of the original feature space. Similarly, label relevance is not considered. MIFS [15] decomposes the multi-label output space into the latent semantic space and metric space of multi-label information. After introducing latent semantic relevance, the decomposed semantic space is used to learn label-specific features. CSFC [16] proposes a convex semi-supervised multi-label learning algorithm with better performance in large-scale processing datasets. And it performs multi-label specific feature selection by incorporating sparsity. The above multi-label learning algorithm can be assumed as a linear mapping from the instance space $\boldsymbol{X}$ to the category label space $\boldsymbol{Y}$ through the coefficient matrix $\boldsymbol{W}$, i.e., a function $f : \boldsymbol{X} \rightarrow \boldsymbol{Y}$ is built for multi-label classification after considering the label relevance. Although these algorithms already show excellent performance, most multi-label learning methods only consider one-way projection from instance space to label space. Obviously, it is a lossy mapping compression that inevitably leads to the loss of discriminative information when mapping instance space to label space [17]. In general, the dimensionality of the instance feature space is much higher than that of the category label space, and this reduction in dimensionality can lead to the loss of useful information in the instance space species during multi-label classification, thus affecting the performance of multi-label classification. To solve this problem, the reverse mapping from feature space to label space ought be considered for multi-label classification. Therefore, this paper proposes a bidirectional loss model that integrates forward mapping loss and reverse reconstruction loss. Specifically, reducing the loss of original information, i.e., forward mapping is the step of learning instance feature-to-category label relationships, and minimizing reconstruction loss denotes the step of reverse mapping to compensate for the loss of information from the previous step.

Moreover, previous research shows that the performance of multi-label classifiers may be improved by exploiting the interrelationships between labels [12, 18]. The typical solution is to consider the pairwise correlations between labels [15, 19]. However, this strong correlation between labels does not capture the causal relationship between them but only indicates their co-occurrence. For example, there may be a strong correlation between eating oranges and treating scurvy because eating oranges can treat scurvy. But it is clear that eating oranges is not a reasonable explanation for scurvy at all. In fact, the rational explanation for treating scurvy is the vitamin C contained in oranges. They are a causal association rather than a simple correlation. Therefore, it has crucial research significance to explore the essential causal relationship between the labels, which makes the model more interpretable [20].

As mentioned above, existing methods may not be able to entirely characterize the information of instance features and effectively utilize the potential causal relationships between labels. Therefore we propose a bi-directional mapping for multi-label learning of label-specific features algorithm that considers both reconstruction loss regularity terms and label causality. Unlike previous multi-label learning models, the proposed algorithm aims to reconstruct the instance feature space from the category label space to accomplish more accurate multi-label classification. As shown in Fig. 1, the reverse mapping $g : \boldsymbol{Y} \rightarrow \boldsymbol{X}$ reconstructs the input instance feature space from the output label space in reverse, which is applied to compensate for the information loss caused by the forward mapping. The label causality matrix characterizes the essential causal relationships between labels. The main contributions of this paper are as follows:

- Combined with the sparse matrix $\boldsymbol{W}$ to extract label-specific features. BDLS can perform label-specific feature extraction while acting as a multi-label classification method.
- A bidirectional loss model combining forward mapping loss and reverse reconstruction loss is proposed. The reconstruction mapping from the category label space to the instance feature space is introduced in the multi-label classification model.
- For the first time, label causality is proposed for representing the essential dependencies between labels and for more accurate multi-label classification.
- Experiments on ten multi-label datasets show that BDLS has better performance than other state-of-the-art algorithms in multi-label classification.

We organize the rest of the paper as follows. Section 2 reviews previous work related to multi-label learning, including multi-label classification and multi-label specific feature learning. Section 3 describes the proposed BDLS

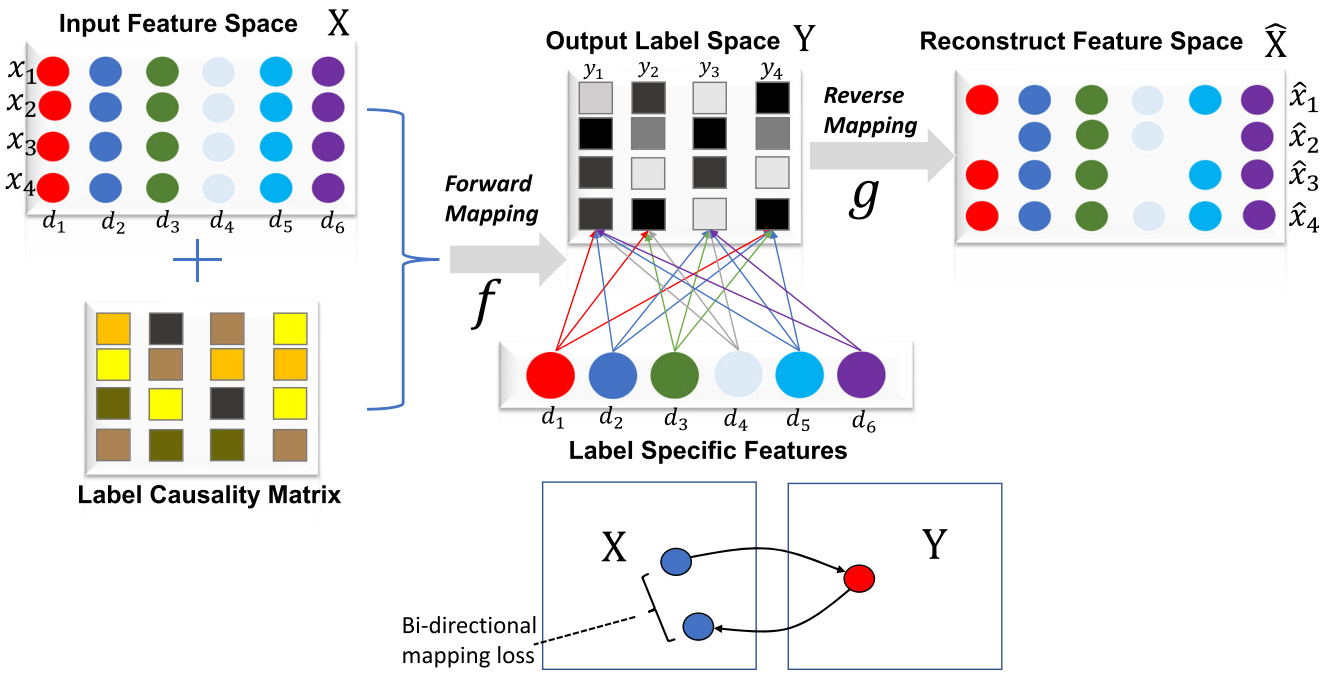


Fig. 1 The learning framework of the proposed method BDLS

algorithm in detail. And the comparison experiments and the analysis of the experimental results will be presented in Section 4. Finally, Section 5 summarizes the paper.

2 Related work

This section describes the work related to multi-label learning algorithms and label-specific feature learning.

2.1 Multi-label learning

Multi-label learning is the method that solves the classification problem of associating a single instance with multiple category labels [2, 11, 21], i.e., learning an efficient function to predict the corresponding set of category labels for invisible instances. In the past decades, many algorithms to deal with multi-label classification have been proposed and widely adopted. By definition in [1], multi-label learning methods can be classified into problem transformation methods and algorithm adaptation methods. The problem transformation methods view a multi-label learning problem as collecting multiple independent single-label learning problems. The typical problem transformation approaches are random k-labelsets (RAKEL) [22], pruning problem transformation (EPS) [23], pairwise comparison ranking method (RPC) [24], calibration marker ranking (CLR) [25], and INSDIF algorithm [26]. Algorithm adaptive methods make appropriate improvements in traditional supervised learning algorithms, such as support vector machines(SVM) [27], decision trees [28], K-nearest neighbor algorithms [10], and AdaBoost algorithms [4]. The representative algorithm adaptive methods is the improved back-propagation multi-label learning algorithm BP-MLL [29]. The algorithm proposes a global error function for multi-label classification using a backpropagation algorithm and gradient descent method to characterize multi-label learning. In addition, an instance-based logistic regression IBLR [30] realizes multi-label classification with optimal balance coefficients between instances and labels by using the labels of neighboring instances as features of the instances for optimal regression estimation.

Meanwhile, the purpose of multi-label learning is to predict the label set of invisible instances by analyzing the training instances of known label sets. Multi-label classification will become extremely challenging when the set of labels is pretty sizable, so the research of label correlation is proposed to solve this problem [11, 31]. By investigating the dependencies between labels, the multi-label learning algorithms proposed in recent years can be classified into three categories [32].

– The first-order strategy treats multi-label learning problems as multiple single-label problems, which obviously ignore the dependencies between labels, e.g., BR [9], ML-KNN [10], LIFT [13]. A prototypical first-order approach is the sparse mapping instance-based multi-label learning method SWIM [33]. It uses the PLS algorithm to mine the prospective sparsified mapping relationships between training and test instances, which

improves robustness while reducing noise interference on the model.

- The second-order strategy performs multi-label learning using pairwise dependencies between labels. One is based on the dependence between two pairs of labels, such as RAKEL [22], ML-TLLT [34], and MLME [35]. The other is to build a model with ranking loss in the objective function, such as Rank-SVM [27], BP-MLL [29], and RELIAB [36]. The classical second-order method ML-TLLT proposes a multi-label teaching and learning algorithm that explores the relationship between possible labels with missing label instances and labels with complete label instances. And it considers the determinism of label features the relevance of paired labels to sufficiently uncover the dependencies between labels for multi-label learning.
- The high-order strategy is a multi-label learning method that fully considers the inherent dependencies of the label set, such as evaluating the dependence of each label on all other labels [30, 37, 38], or considering the interactions between random subsets of the label set [22, 23, 39]. Multi-label learning method LI-MLC [40] based on label inference is a common higher-order multi-label learning approach. It uses label feature selection to explore label relevance. Specifically, LI-MLC applies association rule mining algorithms to find a set of strongly correlated label sets for each instance to extract the corresponding features of the labels. Then it uses the multi-label classifier to predict appropriate labels for invisible instances.

2.2 Label-specific features learning

In multi-label learning, in addition to considering the interrelationships between the labels, the labels may have specific characteristics. Label-specific feature learning has become a significant research direction in recent years. Numerous label-specific feature learning algorithms have emerged in the past decade or so.

LIFT [13] is a typical algorithm for label-specific feature learning. It learns each label-specific feature by clustering each category label into positive and negative instance clusters through the k-means algorithm and calculating the distance between the centers of the clusters and the instances. But the label relevance is not considered. Ml-DFL [41] constructs new label features using a spectral clustering algorithm to explore the similar structure between positive and negative instances. And multi-label classification is performed based on the new label features. LSDM [42] suggests that the samples are first analyzed by positive and negative instance clustering, where the spectral clustering algorithm is spectral instance alignment (SIA). Then multiple label reconstruction feature spaces are constructed based on the clustering results using distance mapping and linear representation. Finally, the information that best expresses the label features is learned from the multiple label reconstruction feature space by simplified linear discriminant analysis techniques (sLDA). Also, label relevance is not considered. MLSF [43] combines meta-label learning with label-specific feature learning. Specifically, it assigns strongly correlated labels together to form k meta-label sets and learns the specific features of these meta-label sets under the condition of considering the relevance of the meta-labels. Meta-label learning in MLSF is a method for analyzing instance space information and label space information through spectral clustering.

Besides, the typical applications in label-specific feature learning also include subset feature selection approaches. LLSF [44] and LLSF-CI [12] learn relevant feature subsets from the original label feature space by sparse superposition for label-specific feature extraction and introduce label correlation to predict better category labels, where LLSF-CI also considers instance correlation. SLMLC [45] proposes a model coefficient matrix combining a sparse matrix for each label-specific feature and a low-rank matrix representing the feature space shared by all labels and performs multi-label classification after considering label correlation. LSML [46], on the other hand, is a multi-label learning algorithm explicitly designed for missing label datasets to learn label-specific features. By learning higher-order label relevance, completes the missing labels and then performs multi-label classification. In the same way, SFUS [14] learns the most relevant features of a label by subspace sparse feature selection and mines the shared subspace of the original feature space to learn label-specific features.

However, label-specific feature learning based on the subset feature selection method described above only considers a one-way projection from the instance space to the label space. That is, learning a coefficient matrix as a specific feature of the label. This method of mapping the instance set to the label set by the coefficient matrix can be regarded as a mapping compression of the data. As can be seen from [17], some instance information may be lost during the mapping process, and the loss of data possibly leads to inaccurate predictive labeling. To deal with this problem, inspired by the self-encoder [47], the BDLS algorithm we proposed in this paper incorporates the loss of reverse reconstruction from label space to instance space while considering the loss of mapping from instance space to label space. Therefore, it reduces the loss of partial information in the instance data when feature selection is performed. Specifically, assuming that $\boldsymbol{X}$ is the instance feature matrix and $\boldsymbol{Y}$ is the category label matrix, the loss of the original data caused by forward mapping can be optimized by establishing the reverse mapping $g : \boldsymbol{X} \rightarrow \boldsymbol{Y}$. That is, minimizing the loss of label

space reverse projection instance space. Unlike previous multi-label learning algorithms, BDLS can rebuild the input instance feature space from the output category label space and incorporate label causality to exploit the essential dependencies between labels more effectively. Thus, BDLS can predict the category labels for invisible instances more accurately than other methods, and experiments on several available datasets fully demonstrate the effectiveness of the proposed reconstruction loss.

3 Proposed approach

This section presents a multi-label learning model with a bi-directional loss function using comprehensive reconstruction constraints. We first describe the reconstruction loss regularization term from feature space to label space, then illustrate the interpretability of label causality, and finally introduce an accelerated proximal gradient optimization method for solving the model.

3.1 Preliminaries

In the case of a multi-label dataset, the matrix of instances with n samples and d features is denoted as $\boldsymbol{X} = [x_1, \cdots, x_n]^T \in \mathbb{R}^{n\times d}$, where $x_i = [x_{i1}, \cdots, x_{id}] \in \mathbb{R}^d$ indicates the feature vector of the instance. The category label matrix is represented as $\boldsymbol{Y} = [y_1, \cdots, y_n]^T \in \mathbb{R}^{n\times l}$, where l represents the number of labels and $y_i = [y_{i1}, \cdots, y_{il}] \in \{0, 1\}^l$ is the label vector. $y_{ij} = 1$ means that the i-th instance x_i has the j-th label y_j, and $y_{ij} = 0$ indicates that the i-th instance x_i does not have the label y_j. With the given multi-label dataset, we intend to learn a coefficient matrix $\boldsymbol{W}$ for multi-label classification that can adequately represent the mapping relationships from instance features to category labels and ultimately achieve multi-label classification.

3.2 Learning label-specific features

For the multi-label datasets, we assume that each category label is identified only with a subset of features of the given dataset. And we construct linear regression models to modify the label-specific features by adding l_1 parametric regularization to ensure the sparsity of the feature or feature sets corresponding to each category label. So the optimization problem can be written as,

$$\min_{\boldsymbol{W}} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{W} - \boldsymbol{Y}\|_F^2 + \beta \|\boldsymbol{W}\|_1 \tag{1}$$

where $\boldsymbol{W} = [w_1, \cdots, w_l] \in \mathbb{R}^{d\times l}$ is the linear regression coefficient, and the parameter $\beta \geq 0$ controls the sparsity of the coefficient matrix $\boldsymbol{W}$. Due to l_1 parametric regularization, some elements of w_i will be zero, and the features corresponding to the zero elements do not contribute to the recognition of the j-th category label $y_j (1 \leq j \leq l)$. In contrast, other features corresponding to non-zero elements of w_i are highly distinguishable for category label y_j. The feature dimensionality of a multi-label dataset can be reduced by selecting label-specific features.

3.3 Bi-directional mapping learning

By learning the label-specific features, we establish a mapping function $f : \boldsymbol{X} \rightarrow \boldsymbol{Y}$ from the instance feature matrix $\boldsymbol{X}$ to the category label matrix $\boldsymbol{Y}$, where $\boldsymbol{W}$ is the mapping coefficient matrix for encoding the label-specific features. Generally, the loss of discriminative information during the mapping process may be caused by the fact that the dimensionality of the input feature matrix is much higher than that of the output label matrix, i.e., $d \gg l$ [17]. Therefore, by introducing the reconstruction loss function to reconstruct the input feature matrix from the output label matrix to reduce the loss of discriminative information and make the predicted label matrix similar to the original label matrix. Then the objective function can be extended as,

$$\min_{\boldsymbol{W}} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{W} - \boldsymbol{Y}\|_F^2 + \frac{\lambda}{2} \left\|\boldsymbol{Y}\boldsymbol{W}^{\mathrm{T}} - \boldsymbol{X}\right\|_F^2 + \beta \|\boldsymbol{W}\|_1 \tag{2}$$

where λ is a parameter, which balances the relative importance of reconstruction loss on $\boldsymbol{W}$. By introducing the reconstruction loss regularization term, it can be used to measure the similarity between the input feature matrix and the reconstruction feature matrix in the output label matrix. Specifically, the reverse mapping function $g : \boldsymbol{Y} \rightarrow \boldsymbol{X}$ is established to exploit the intrinsic relationship between instances and labels fully. Thus, two objectives can be achieved by (2):

1. Since l_1-norm regular term ensures the sparsity of $\boldsymbol{W}$ to extract label-specific features.
2. In considering the mapping loss for label-specific feature learning, the reconstruction loss regularization term is added to compensate for the loss of discriminative information during the mapping process.

3.4 Exploiting label causality

Previous research results indicate that the performance of multi-label classifiers can be improved by exploiting the interrelationships between labels [19, 46]. That is, utilization of intrinsic relationships between labels can better solve multi-label classification issues. The inherent relationship between the labels is not considered in (2). Therefore, in this paper, we propose a label causality to mine the essential connection between labels. Specifically,

Causal Learner [48] with the GSBN algorithm [49], a global causal structure learning method, is applied to explore label causality. First, it gets the Markov Blanket(MB) or Parents and Children(PC) for each label by partial-to-whole structure learning, then uses the learned MB or PC to construct the directed acyclic graph (DAG) framework, and finally obtains the global causal structure by determining the causal direction through score-based or constraint-based causal learning methods [50].

Under the constraint of considering label causality, let $\boldsymbol{A} \in \mathbb{R}^{l\times l}$ be the label causality matrix, and a_{ij} denote the causal relationship between label y_i and label y_j. We learn the specific features of the labels by computing the Euclidean distance $a_{ij}\left\|w_i - w_j\right\|_2^2$ between the coefficient vectors w_i and w_j. This method indicates that when y_i and y_j are causally related, their characteristics will be similar, and the corresponding coefficient w_i and w_j will be more close to each other; conversely, the similarity may be weak. After introducing label causality, the objective function can be optimized as,

$$\min_{\boldsymbol{W}} \frac{1}{2}\|\boldsymbol{XW}-\boldsymbol{Y}\|_F^2+\frac{\lambda}{2}\left\|\boldsymbol{YW}^{\mathrm{T}}-\boldsymbol{X}\right\|_F^2+\alpha\mathrm{tr}(\boldsymbol{WAW}^{\mathrm{T}})+\beta\|\boldsymbol{W}\|_1 \quad (3)$$

where α is the trade-off coefficient to balance the importance of label causality on $\boldsymbol{W}$, and the causality matrix $\boldsymbol{A}$ is defined as follows:

$$a_{ij}=\begin{cases}1 & y_i\to y_j\\ 0 & y_i\nrightarrow y_j\end{cases}\ (1\le i\le l, 1\le j\le l) \quad (4)$$

where $y_i \to y_j$ indicates that y_i to y_j is a cause-to-effect relationship, then $a_{ij} = 1$, and conversely $y_i \nrightarrow y_j$ indicates that y_i to y_j is not a cause-to-effect relationship, then $a_{ij} = 0$. Specifically, the label causality matrix **A** is computed by the Causal Learner and it can be represented visually by the directed acyclic graph. As shown in Fig. 2, the directed acyclic graph of the label causality matrix A obtained using Causal Learner on the Emotion dataset. Nodes ① to ⑥ in Fig. 2 correspond to all category labels of the Emotion dataset, and the directed line segments between the nodes indicate the cause-to-effect association between the labels, which is expressed as $y_i \to y_j$ in (4).

As mentioned above, a bidirectional mapping multi-label learning framework that considers label causality is shown in (3). The specific features of the labels are obtained by learning the optimization parameters **W**. The second and third terms are the novel reconstruction loss regularity term and label causality regularity term proposed in this paper. The reconstruction loss regular term can balance the loss of discriminative information in the original feature space due to forward mapping. The label causality regular term is used to learn the essential relationship between labels for the purpose of optimizing the parameter **W**.We use the accelerated proximal gradient method to optimize (3). The specific optimization steps will be given in the next subsection.

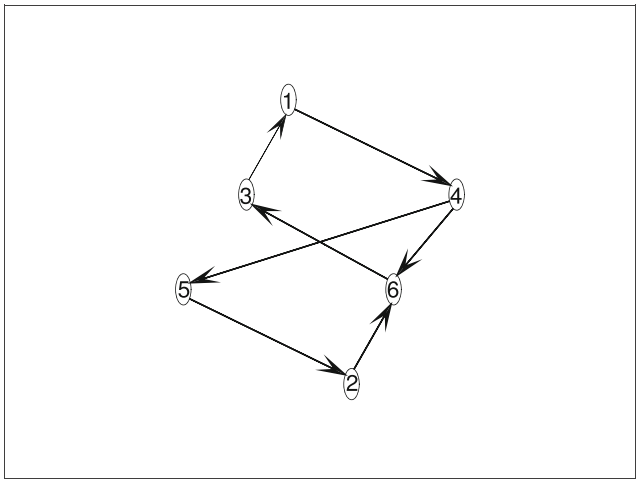


Fig. 2 Label causality directed acyclic graph of Emotion dataset. Nodes ①, ②, ③, ④, ⑤, ⑥ represent the category labels of the Emotion dataset, and the directed line segments pointing from one node to another node represent the causal association between the labels

3.5 Optimization

The minimization problem of (3) is a non-smooth convex optimization problem. Because of the non-smoothness of the l_1 regularization, the accelerated proximal gradient method can be applied to seek this non-smooth optimization problem [51]. Typically, the accelerated proximal gradient method can be written as the following convex optimization problem,

$$\min_{\Phi\in\mathrm{H}}\{F(\Phi)=f(\Phi)+g(\Phi)\} \quad (5)$$

where H is the real Hilbert space, $f(\Phi)$ is a smooth convex function, $g(\Phi)$ is usually a non-smooth convex function, and $f(\Phi)$ is further Lipschitz continuous. Thus, we can obtain $\|\nabla f(\Phi_1)-\nabla f(\Phi_2)\|\le L_f\|\Delta\Phi\|$, where $\Delta\Phi = \Phi_1 - \Phi_2$, and L_f is the Lipschitz constant. Instead of directly minimizing function $F(\Phi)$, the accelerated nearest neighbor algorithm minimizes the separable quadratic approximation sequence $Q_{L_f}\left(\Phi,\Phi^{(k)}\right)$of the $F(\Phi)$, denoted as,

$$Q_{L_f}\left(\Phi,\Phi^{(k)}\right)=f(\Phi^{(k)})+\left\langle\nabla f(\Phi^{(k)}),\Phi-\Phi^{(k)}\right\rangle+\frac{L_f}{2}\left\|\Phi-\Phi^{(k)}\right\|_F^2+g(\Phi) \quad (6)$$

where $\Phi^{(k)} = \Phi_k + \frac{c_{k-1}-1}{c_k}(\Phi_k - \Phi_{k-1})$, Φ_k and Φ_{k-1} are the coefficient matrices of the k and $k-1$ iterations, respectively, and the series c_k should satisfy $c_{k+1}^2 - c_{k+1} \le c_k^2$. In [51], the work has shown that the convergence rate can be increased to $O(k^{-2})$ by this setting. Similarly, according to [51], (6) admits a unique minimizer

$$\Phi^* = \arg\min\{Q_{L_f}(\Phi,\Phi^{(k)}) : \Phi\in\mathrm{H}\}. \quad (7)$$

Simple algebra shows that (ignoring constant terms in $\Phi^{(k)}$)

$$\Phi^* = \arg\min_{\Phi} \left\{ g(\Phi) + \frac{L_f}{2} \left\| \Phi - \left(\Phi^{(k)} - \frac{1}{L_f} \nabla f(\Phi^{(k)}) \right) \right\|_F^2 \right\}. \quad (8)$$

When making

$$G^{(k)} = \Phi^{(k)} - \frac{1}{L_f} \nabla f(\Phi^{(k)}) \quad (9)$$

the solution process of Φ can be written as,

$$\Phi^* = \arg\min_{\Phi} Q_{L_f}(\Phi, \Phi^{(k)}) = \arg\min_{\Phi} g(\Phi) + \frac{L_f}{2} \left\| \Phi - G^{(k)} \right\|_F^2 \quad (10)$$

According to (3) and (5), $f(\Phi)$ and $g(\Phi)$ are defined as follows,

$$f_{\boldsymbol{W}}(\Phi) = \frac{1}{2} \|\boldsymbol{X}\boldsymbol{W} - \boldsymbol{Y}\|_F^2 + \frac{\lambda}{2} \left\| \boldsymbol{Y}\boldsymbol{W}^{\mathrm{T}} - \boldsymbol{X} \right\|_F^2 + \alpha Tr\left(\boldsymbol{W}\boldsymbol{A}\boldsymbol{W}^{\mathrm{T}} \right) \quad (11)$$

$$g_{\boldsymbol{W}}(\Phi) = \beta \|\boldsymbol{W}\|_1 \quad (12)$$

The gradient of W in (11) can be described as,

$$\frac{\partial f_{\boldsymbol{W}}(\Phi)}{\partial \boldsymbol{W}} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W} + \lambda \boldsymbol{W}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} - (1+\lambda)\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y} + \alpha \boldsymbol{W}(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}}) \quad (13)$$

Then, according to (10), (11), and (12), $\boldsymbol{W}$ can be optimized as,

$$\begin{aligned} \boldsymbol{W} &= \arg\min_{\boldsymbol{W}} Q_{L_f}(\boldsymbol{W}, \boldsymbol{W}^{(k)}) \\ &= \arg\min_{\boldsymbol{W}} \frac{L_f}{2} \left\| \boldsymbol{W} - \boldsymbol{G}^{(k)} \right\|_F^2 + g(\boldsymbol{W}) \\ &= \arg\min_{\boldsymbol{W}} \frac{1}{2} \left\| \boldsymbol{W} - \boldsymbol{G}^{(k)} \right\|_F^2 + \frac{\beta}{L_f} \|\boldsymbol{W}\|_1 \end{aligned} \quad (14)$$

where

$$\boldsymbol{G}^{(k)} = \boldsymbol{W}^{(k)} - \frac{1}{L_f} \nabla f(\boldsymbol{W}^{(k)}) \quad (15)$$

Then $\boldsymbol{W}$ is updated as follows,

$$\boldsymbol{W}^{(k)} = \boldsymbol{W}_k + \frac{c_{k-1} - 1}{c_k} (\boldsymbol{W}_k - \boldsymbol{W}_{k-1}) \quad (16)$$

$$\boldsymbol{W}_{k+1} = prox_{\varepsilon} \left(\boldsymbol{W}^{(k)} - \frac{1}{L_f} \frac{\partial f_{\boldsymbol{W}}(\Phi)}{\partial \boldsymbol{W}^{(k)}} \right) \quad (17)$$

where ε is the step size, and the value in this paper is $\frac{\beta}{L_f}$. The soft-threshold operator related to the l_1-norm is defined as,

$$prox_{\varepsilon}(w_{ij}) = (|w_{ij}| - \varepsilon)_+ sign(w_{ij}) \quad (18)$$

where w_{ij} is an element of $\boldsymbol{W}$, and $(\cdot)_+ = \max(\cdot, 0)$, $1 \le i \le d$, $1 \le j \le l$. According to (13), given $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$, we can obtain,

$$\begin{aligned} \|\nabla f(\boldsymbol{W}_1) - \nabla f(\boldsymbol{W}_2)\|_F^2 &= \left\| \boldsymbol{X}^T\boldsymbol{X}\Delta\boldsymbol{W} + \lambda\Delta\boldsymbol{W}\boldsymbol{Y}^T\boldsymbol{Y} + \alpha\Delta\boldsymbol{W}(\boldsymbol{A} + \boldsymbol{A}^T) \right\|_F^2 \\ &\le 2 \left\| \boldsymbol{X}^T\boldsymbol{X}\Delta\boldsymbol{W} \right\|_F^2 + 2 \left\| \lambda\Delta\boldsymbol{W}\boldsymbol{Y}^T\boldsymbol{Y} \right\|_F^2 \\ &\quad + 2 \left\| \alpha\Delta\boldsymbol{W}(\boldsymbol{A} + \boldsymbol{A}^T) \right\|_F^2 \\ &\le 2 \left\| \boldsymbol{X}^T\boldsymbol{X} \right\|_2^2 \|\Delta\boldsymbol{W}\|_F^2 + 2 \left\| \lambda\boldsymbol{Y}^T\boldsymbol{Y} \right\|_2^2 \|\Delta\boldsymbol{W}\|_F^2 \\ &\quad + 2 \left\| \alpha(\boldsymbol{A} + \boldsymbol{A}^T) \right\|_2^2 \|\Delta\boldsymbol{W}\|_F^2 \\ &= \left(2 \left\| \boldsymbol{X}^T\boldsymbol{X} \right\|_2^2 + 2 \left\| \lambda\boldsymbol{Y}^T\boldsymbol{Y} \right\|_2^2 \right. \\ &\quad \left. + 2 \left\| \alpha(\boldsymbol{A} + \boldsymbol{A}^T) \right\|_2^2 \right) \|\Delta\boldsymbol{W}\|_F^2 \\ &= \left(2\sigma_{\max}^2(\boldsymbol{X}^T\boldsymbol{X}) + 2\sigma_{\max}^2(\lambda\boldsymbol{Y}^T\boldsymbol{Y}) \right. \\ &\quad \left. + 2\sigma_{\max}^2(\alpha(\boldsymbol{A} + \boldsymbol{A}^T)) \right) \|\Delta\boldsymbol{W}\|_F^2 \end{aligned} \quad (19)$$

where $\Delta\boldsymbol{W} = \boldsymbol{W}_1 - \boldsymbol{W}_2$, $\sigma_{\max}(\cdot)$ is calculated as the maximum singular value function, From (19) can be obtained,

$$\begin{aligned} \|\nabla f(\boldsymbol{W}_1) - \nabla f(\boldsymbol{W}_2)\|_F^2 \le \left(2\sigma_{\max}^2(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}) + 2\sigma_{\max}^2(\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}) \right. \\ \left. + 2\sigma_{\max}^2(\alpha(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}})) \right) \|\Delta\boldsymbol{W}\|_F^2 \end{aligned} \quad (20)$$

The Lipschitz constant can be found from (20) as,

$$L_f = \sqrt{ \left(2\sigma_{\max}^2(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}) + 2\sigma_{\max}^2(\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}) + 2\sigma_{\max}^2(\alpha(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}})) \right) } \quad (21)$$

Algorithm 1 summarizes the optimization steps of the proposed BDLS based on the accelerated proximal gradient method. By inputting the training data $\boldsymbol{X}$, the label matrix $\boldsymbol{Y}$ and parameter optimization we can obtain the desired model coefficient matrix $\boldsymbol{W}$ which can be utilized to predict the possible labels for unknown instances. Specifically, given a test dataset $\boldsymbol{X}_{test}$, and the predicted labels can be obtained by $sign(\boldsymbol{S}_{test} - \tau)$ based on a given threshold τ, where $\boldsymbol{S}_{test} = \boldsymbol{X}_{test}\boldsymbol{W}$. Algorithm 2 provides the test details of the BDLS.

Algorithm 1 Optimization of BDLS.

Input: Training data matrix: $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, label matrix: $\boldsymbol{Y} \in \mathbb{R}^{n \times l}$, Trade-off parameters: α, β, λ, and γ, Number of iterations: t;

Output: Model coefficient matrix: $\boldsymbol{W} \in \mathbb{R}^{d \times l}$;

Initialization:

$c_0, c_1 \leftarrow 1, k \leftarrow 1, \boldsymbol{W}_0, \boldsymbol{W}_1 \leftarrow (\boldsymbol{X}^T\boldsymbol{X} + \gamma I)^{-1}\boldsymbol{X}^T\boldsymbol{Y}$;

compute L_f according to (21);

compute label causality matrix $\boldsymbol{A}$ using Causal Learner;

while *not converged* **do**

  Update $\boldsymbol{W}^{(k)}$ by (16) : $\boldsymbol{W}^{(k)} \leftarrow \boldsymbol{W}_k + \frac{c_{k-1}-1}{c_k}(\boldsymbol{W}_k - \boldsymbol{W}_{k-1})$;

  Update $\boldsymbol{G}^{(k)}$ by (15) : $\boldsymbol{G}^{(k)} \leftarrow \boldsymbol{W}^{(k)} - \frac{1}{L_f}\nabla f(\boldsymbol{W}^{(k)})$;

  Update $\boldsymbol{W}_{k+1}$ by (17): $\boldsymbol{W}_{k+1} \leftarrow prox_{\frac{\beta}{L_f}}\left(\boldsymbol{G}^{(k)}\right)$;

  Update step size : $c_{k+1} \leftarrow \frac{1+\sqrt{4c_k^2+1}}{2}$;

  $k \leftarrow k + 1$;

$\boldsymbol{W} \leftarrow \boldsymbol{W}_k$;

Algorithm 2 Test of BDLS.

Input: Testing data matrix: $\boldsymbol{X}_{test} \in \mathbb{R}^{m\times d}$, model coefficient matrix: $\boldsymbol{W} \in \mathbb{R}^{d\times l}$, threshold: τ;
Output: Predictive label matrix: $\boldsymbol{Y}_{test}$, score matrix: $\boldsymbol{S}_{test}$;

$\boldsymbol{S}_{test} \leftarrow \boldsymbol{X}_{test}W$;
$\boldsymbol{Y}_{test} \leftarrow sign(\boldsymbol{S}_{test} - \tau)$.

BDLS can also be adopted as a multi-label feature selection method. The attributes corresponding to the non-zero components of w_i can be recognized as specific features of y_i. Therefore, it is available to combine binary classifiers, like BSVM [9], CC [39], LIFT [13] for multi-label classification after label-specific feature learning. The BDLS-SVM algorithm in this paper incorporates the binary classifier BSVM. Algorithm 3 recapitulates the BDLS-SVM algorithm. Specifically, Algorithm 3 uses the method proposed in [9] to train the output coefficient matrix $\boldsymbol{W}$ obtained by Algorithm 1, the training data $\boldsymbol{X}$, and the label matrix $\boldsymbol{Y}$. Finally, the final classification result $\boldsymbol{Y}_{test}$ and a score matrix $\boldsymbol{S}_{test}$ are obtained by testing.

Algorithm 3 BDLS-SVM method.

Input: Training data matrix: $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, label matrix: $\boldsymbol{Y} \in \mathbb{R}^{n\times l}$, Testing data matrix: $\boldsymbol{X}_{test} \in \mathbb{R}^{m\times d}$, Binary classifier: SVM, Kernel function θ: linear;
Output: Predictive label matrix: $\boldsymbol{Y}_{test}$, score matrix: $\boldsymbol{S}_{test}$;

Training:
Learning the model coefficient matrix $\boldsymbol{W}$ of BDLS by Algorithm 1;
for $j = 1$ *to* l **do**
  $d_j \leftarrow find(\boldsymbol{W}_j \neq 0)$;
  $\boldsymbol{X}^j \leftarrow \boldsymbol{X}(:d_j)$;
  $h_j \leftarrow f(\boldsymbol{X}^j, \boldsymbol{Y}, \theta)$;
testing:
for $j = 1$ *to* l **do**
  $\boldsymbol{X}^j_{test} \leftarrow \boldsymbol{X}_{test}(:d_j)$;
  $\{\boldsymbol{Y}^j_{test}, \boldsymbol{S}^j_{test}\} \leftarrow h_j(\boldsymbol{X}_{test}, \boldsymbol{Y}_{test}, \theta)$;
$\boldsymbol{Y}_{test} \leftarrow \left[\boldsymbol{Y}^1_{\text{test}}, \boldsymbol{Y}^2_{\text{test}} \ldots, \boldsymbol{Y}^l_{\text{test}}\right]$;
$\boldsymbol{S}_{test} \leftarrow \left[\boldsymbol{S}^1_{\text{test}}, \boldsymbol{S}^2_{\text{test}} \ldots, \boldsymbol{S}^l_{\text{test}}\right]$;

3.6 Complexity analysis

The time complexity of BDLS depends strongly on matrix multiplication operations. The size of each matrix of Algorithm 1: $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, $\boldsymbol{Y} \in \mathbb{R}^{n\times l}$, $\boldsymbol{W} \in \mathbb{R}^{d\times l}$, $\boldsymbol{A} \in \mathbb{R}^{l\times l}$. The step 1 initialization process occurs only once, so the time complexity mainly depends on the iterative process when the algorithm does not converge. Specifically, $\boldsymbol{X}^T\boldsymbol{X}$, $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}$, and $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}$ are computed at the initialization process, so the complexity of computing in step 7 is $O(d^2l + ndl + dl^2)$. Thus the total time complexity can be written as $O((n + d + l)ldt)$, where t is the number of iterations at convergence. In the experiment, t usually does not exceed 80 times. Generally speaking, the time complexity of BDLS is proportional to the number of samples when the dataset satisfies $n \gg d > l$. Moreover, due to the sparsity of $\boldsymbol{W}$, the actual time cost is reduced. Meanwhile, in this paper, the complexity of the BDLS algorithm is also compared with the LLSF, LSML, LSF-CI, and JLCLS algorithm. The specific information about these four algorithms will be given in Section 4.2. According to the works in [44], [46], [12], [52], it is known that the complexity of LLSF is $O(d^2 + dl + l^2 + nd + nl)$, the complexity of LSML is $O\left((n + l)d^2 + (n + d)l^2 + dnl + l^3 + d^3\right)$, the complexity of LSF-CI is $O\left((n + l)d^2 + ndl + dl^2 + n^2d\right)$, and the complexity of JLCLS is $O((n + 1)(d^2l^2 + nl^2 + nd^2l) + d^3 + l^3)$. The comparison shows that the complexity of the BDLS algorithm is competitive with the other four algorithms, which further illustrates the effectiveness of BDLS.

4 Experiment

In this section, we conduct extensive experiments on ten real-world datasets to illustrate the effectiveness of our proposed approach.

4.1 Data sets

The ten datasets used in the experiments are publicly available, and all of them can be downloaded in Mulan[1]. The statistics are summarized in Table 1, where $\boldsymbol{n}$ is the number of instances, $\boldsymbol{d}$ is the number of instance features, $\boldsymbol{l}$ is the number of category labels, $\boldsymbol{C}$ is the average number of labels, and $\boldsymbol{D}$ is the domain to which the dataset belongs.

4.2 Evaluation metrics

We select six general evaluation metrics: Hamming Loss, Average Precision, One-Error, Coverage, Ranking Loss, and macro-averaged AUC [2, 53], to validate the model performance and compare them under different algorithms. For convenience, the six evaluation metrics are abbreviated as HL ↓, AP ↑, OE ↓, CV ↓, RL ↓, and AUC ↑, where ↑ indicates better for higher values and ↓ shows better for lower values. Given a test dataset $D_t = \{(x_i, Y_i)\}_{i=1}^{n_t}$,

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

$h(x_i)$ is the multi-label classifier, $f(x_i, y)$ is the prediction function, and $rank_f$ is the ranking function. The specific evaluation metrics are as follows.

- **Hamming Loss (HL ↓):** Calculate the number of instance-label pairs that are misclassified, indicating the number of false matches between the actual and predicted labels of the instance sample.

$$\mathrm{HL}_D(h) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y|}|h(x_i) \neq Y_i| \tag{22}$$

- **Average Precision (AP ↑):** Calculate the average score of the correct labels aligned by a specific label $y \in y_i$.

$$AP_D(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|}\sum_{y\in Y_i}\frac{\left|y'|rank(x_i, y') \leq rank_f(x_i, y), y' \in Y_i\right|}{rank_f(x_i, y)} \tag{23}$$

- **One-Error (OE ↓):** Calculate the score of instances where the highest-ranked labels are not in the set of related labels.

$$OE_D(f) = \frac{1}{n}\sum_{i=1}^{n}[[\arg\max_{y\in Y} f(x_i, y)] \notin Y_i] \tag{24}$$

- **Ranking Loss (RL ↓):** Metrics for evaluating the ranking of irrelevant labels higher than that of related labels.

$$\begin{aligned} RL_D(f) = & \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|\left|\bar{Y}_i\right|}|\{(y_1, y_2)|f(x_i, y_1) \\ & \leq f(x_i, y_2), (y_1, y_2) \in Y_i \times \bar{Y}_i\}| \end{aligned} \tag{25}$$

- **Coverage (CV ↓):** Indicators for evaluating the average number of steps to traverse all relevant labels for a given sample.

$$CV_D(f) = \frac{1}{n}\sum_{i=1}^{n}\max_{y\in Yi} rank_f(x_i, y) - 1 \tag{26}$$

- **Macro-averaged AUC (AUC ↑):** Evaluate the average AUC across all categories of labels.

$$AUC_{\mathrm{macro}} = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|\left\{\left(x', x''\right) \mid f_i\left(x'\right) \geq f_i\left(x''\right), \left(x', x''\right) \in \mathrm{P}_i \times \mathrm{N}_i\right\}\right|}{|\mathrm{P}_i|\,|\mathrm{N}_i|} \tag{27}$$

where $P_i = \{x_j|y_i \in Y_j, 1 \leq j \leq p\}$ denotes the set of test instances with the label y_i, $N_i = \{x_j|y_i \notin Y_j, 1 \leq j \leq p\}$ represents the set of test instances without the label y_i.

4.3 Comparative algorithms

For validating the effectiveness of our proposed model, we compare it with the following state-of-the-art multi-label learning methods:

- **LLSF**[2] [44]: A multi-label learning method for label-specific features considering label correlation. The index intervals of parameters α and β are $\{2^{-10}, 2^{-9}, \cdots, 2^9, 2^{10}\}$, and the range of values of η is $\{0.01, 0.1, 1, 10\}$. The threshold τ is taken as 0.5.
- **LIFT**[3] [13]: A multi-label learning method with label-specific features. It performs a clustering analysis of positive and negative instances for each label, constructs specific features, and makes a classification based on the clustering results. The parameter r is set to 0.1.
- **ML-kNN**[4] [10]: A lazy learning method for multi-label learning. It learns the k nearest neighbor instances of the invisible instances in the dataset and counts the information obtained from the label set of these neighboring instances. The maximum posterior probability is exploited to predict the possible labels of the instance. And the nearest neighbor number in the experiment $K = 10$.
- **LSML**[5] [46]: Learning label-specific features for improved label-deficient multi-label classification. Learning higher-order label relevance complements the missing label matrix to augment the new label matrix. And then, it learns label-specific representations to perform multi-label classification. All the parameters of the method are tuned in$\{10^{-5}, 10^{-4}, \cdots, 10^2, 10^3\}$.
- **LSF-CI** [12]: A multi-label learning method for learning label-specific features using relevant information. It performs multi-label classification by considering both instance-related information in the feature space and label-related information in the label space, in which instance-related information is calculated using the probabilistic neighborhood graph model and label-related information is calculated using the cosine similarity. The parameters α, β select from $\{2^{-10}, 2^{-9}, \cdots, 2^9, 2^{10}\}$, and the parameter γ selects from $\{2^{-12}, 2^{-11}, \cdots, 2^{11}, 2^{12}\}$. The threshold τ is taken as 0.5.
- **JLCLS** [52]: A multi-label learning framework using joint label complementation and label-specific features. The algorithm uses an alternating iteration method to obtain the completion matrix and label-specific features based on sufficient consideration of label

[2] code:http://www.escience.cn/people/huangjun/index.html.

[3] code:http://palm.seu.edu.cn/zhangml.

[4] code:http://palm.seu.edu.cn/zhangml.

[5] code:http://www.escience.cn/people/huangjun/index.html.

Table 1 Statistics of the ten datasets

Datasets	n	d	l	C	D
Arts	5000	462	26	1.636	Text
Computers	5000	681	33	1.508	Text
Emotion	593	72	6	1.869	Music
Business	5000	438	30	1.588	Text
Social	5000	1047	39	1.233	Text
Entertainment	5000	640	21	1.420	Text
Medical	978	1449	45	1.245	Text
Cal500	502	68	174	26.044	Music
Society	5000	636	27	1.461	Text
Recreation	5000	606	22	1.423	Text

relevance. The parameters α, β and θ select from $\{2^{-10}, 2^{-9}, \cdots, 2^{9}, 2^{10}\}$, and the parameter γ selects from $\{0.1, 1, 10\}$.

- **BDLS**: The method proposed in this paper. By introducing reconstruction mapping loss, reducing the loss of discriminative information caused by dimensionality drop. The index intervals of parameters α, λ, and β are $\{2^{-7}, 2^{-6}, \cdots, 2^{6}, 2^{7}\}$, and the range of values of γ is $\{0.01, 0.1, 1, 10\}$. The threshold τ is taken as 0.5.
- **BDLS-SVM**: A binary classifier SVM is added to BDLS for multi-label classification. All parameters are the same as BDLS settings. The kernel function of SVM is linear.

On all algorithms, the datasets are experimented with a five-fold cross-validation to avoid problems caused by unreasonable division of the datasets. Furthermore, as mentioned above, each comparison algorithm selects the optimal parameter values according to the parameter ranges provided in the original papers. All experiments are repeated 10 times and the means and standard deviations are reported. For BDLS and BDLS-SVM source code is publicly available at https://github.com/Tanyi003/BDLScode.

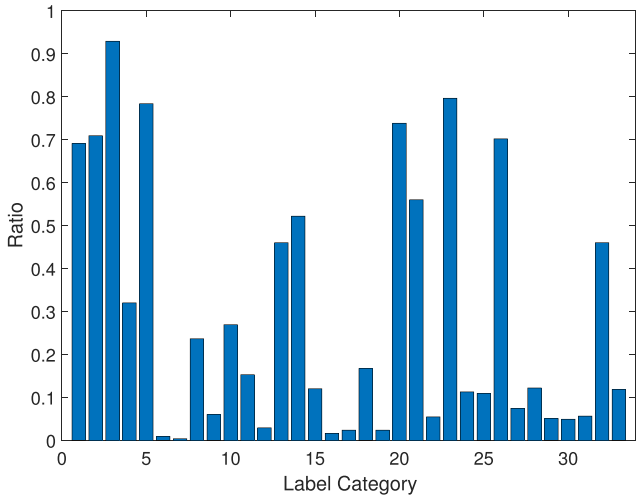
4.4 Experimental results

4.4.1 Visualization of label-specific features

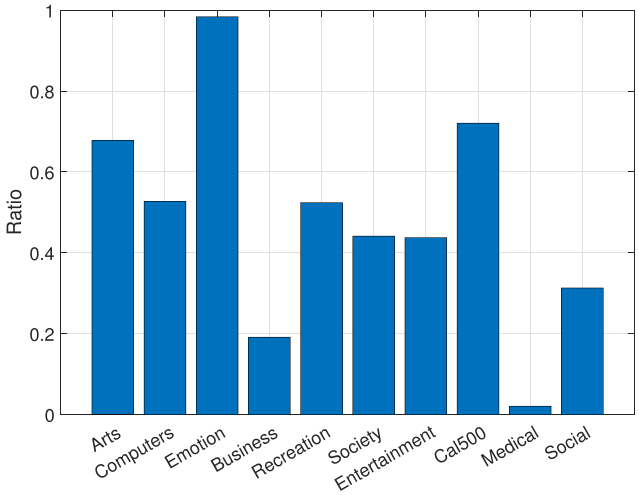
We do some relevant experiments on the Education dataset to evaluate the effectiveness of BDLS in extracting label-specific features. The proportion of instances corresponding to the category labels on the Education dataset is shown in Fig. 3a. From Fig. 3a, the features corresponding to each category label are different and smaller than the original number of instance features. It can achieve the goal of feature dimensionality reduction and extract unique features corresponding to the labels. And the sparsity of $\boldsymbol{W}$ is designed for this purpose. Figure 3b shows the ratio of feature dimensionality reduction on each dataset, which is the ratio of the number of features corresponding to the labels to the total number of features in the instance. It can be seen that label-specific feature learning can significantly reduce the feature dimensionality and improve the performance of algorithm.

4.4.2 Results of multi-label classification

In addition to the exploration of the validity of label-specific features, we compare the results of each dataset under different algorithms. A five-fold cross-validation was used for the experiments and the means and standard deviations obtained from ten repetitions of each algorithm on ten datasets are reported in Tables 2, 3, 4, 5, 6 and 7, where the best experimental results are expressed in bold. From Tables 2-7, it can be seen that BDLS-SVM significantly outperforms the LSML, KNN, and JLCLS algorithms in most cases for HL, AP, and RL. In terms of HL metrics, BDLS-SVM shows the best classification performance on the eight datasets, but slightly inferior to LIFT and LSF-CI on Emotion and Cal500. In a similar way to HL, the proposed method performs best on 8 datasets for both AP as well as CV metrics. And for the RL metric, BDLS-SVM shows the best results on all nine datasets. However, for the OE and AUC metrics, the proposed method showed suboptimal results on three and four datasets, respectively. It can also be noted that the BDLS as well as BDLS-SVM are not optimal for all six metrics on the Cal500 dataset. The most important reason is that the number of instance feature dimensions of Cal500 is much smaller than the number of labels, and the reconstruction loss canonical term of the bidirectional mapping framework increases the loss of feature space discriminative information and makes the classification performance worse. Finally, by comparing the BDLS-SVM and LIFT algorithms that use binary



(a) Feature ratio on Education



(b) Feature ratio on ten datasets

Fig. 3 Ratio of label-specific features on Education and ratio of label-specific features on all datasets

classifiers, we notice that the BDLS-SVM outperforms AP, HL, CV and OE metrics on most of the datasets, and is comparable to LIFT in RL and AUC. Based on the overall consideration, the proposed method in this paper has achieved effective multi-label classification.

4.4.3 Statistical hypothesis test

Furthermore, Friedman test [54] is utilized to compare the performance of various algorithms more effectively. Friedman statistics F_F and the critical values of each evaluation index are given in Table 8. As shown in Table 8, the hypothesis that all algorithms perform equally is explicitly rejected at the significance level $\alpha = 0.05$. Therefore, Nemenyi Test [54] is performed to test the

Table 2 The results (mean±std) of all 10 datasets in Hamming Loss (HL ↓)

Datasets	BDLS	BDLS -SVM	LLSF	LSML	KNN	LIFT	LSF-CI	JLCLS
Arts	0.0559±0.0010	**0.0522±0.0007**	0.0580±0.0009	0.0664±0.0017	0.0599±0.0009	0.0529±0.0016	0.0535±0.0016	0.0558±0.0013
Computers	0.0381±0.0012	**0.0325±0.0012**	0.0378±0.0018	0.0386±0.0007	0.0386±0.0014	0.0330±0.0006	0.0341±0.0017	0.0359±0.0009
emotion	0.1880±0.0086	0.1978±0.0111	0.1982±0.0039	0.2015±0.0089	0.2020±0.0120	**0.1869±0.0038**	0.1975±0.0220	0.2583±0.0068
Business	0.0277±0.0007	**0.0241±0.0011**	0.0513±0.0076	0.0249±0.0009	0.0265±0.0015	0.0244±0.0009	0.0268±0.0012	0.0280±0.0007
Recreation	0.0554±0.0014	**0.0520±0.0017**	0.0552±0.0006	0.0677±0.0021	0.0602±0.0012	0.0529±0.0022	0.0535±0.0017	0.0573±0.0008
Society	0.0548±0.0008	**0.0510±0.0017**	0.0567±0.0014	0.0647±0.0027	0.0545±0.0018	0.0511±0.0013	0.0514±0.0020	0.0533±0.0009
Entertainment	0.0567±0.0008	**0.0493±0.0015**	0.0541±0.0009	0.0629±0.0047	0.0621±0.0012	0.0500±0.0040	0.0513±0.0024	0.0549±0.0007
Medical	0.0106±0.0005	**0.0097±0.0013**	0.0103±0.0010	0.0110±0.0012	0.0156±0.0006	0.0125±0.0005	0.0100±0.0016	0.0192±0.0010
Cal500	0.1396±0.0021	0.1377±0.0047	0.1384±0.0010	0.1973±0.0105	0.1397±0.0038	0.1380±0.0028	**0.1370±0.0030**	0.1370±0.0030
Social	0.0256±0.0007	**0.0189±0.0005**	0.0356±0.0078	0.0240±0.0012	0.0210±0.0007	0.0195±0.0006	0.0204±0.0010	0.0218±0.0005

Table 3 The results (mean±std) of all 10 datasets in Average Precision (AP ↑)

Datasets	BDLS	BDLS -SVM	LLSF	LSML	KNN	LIFT	LSF-CI	JLCLS
Arts	0.6233±0.0098	**0.6390±0.0131**	0.6085±0.0081	0.6286±0.0102	0.5267±0.0093	0.6260±0.0041	0.6228±0.0156	0.6299±0.0083
Computers	0.6861±0.0097	**0.7176±0.0072**	0.6926±0.0110	0.7132±0.0084	0.6446±0.0156	0.7076±0.0094	0.7059±0.0103	0.7048±0.0055
emotion	0.8129±0.0137	**0.8186±0.0101**	0.8079±0.0092	0.8022±0.0118	0.7965±0.0116	0.8148±0.0136	0.8121±0.0370	0.7624±0.0075
Business	0.8774±0.0066	**0.8949±0.0071**	0.8645±0.0061	0.8877±0.0052	0.8831±0.0123	0.8925±0.0049	0.8768±0.0120	0.8861±0.0045
Recreation	0.6383±0.0108	**0.6519±0.0017**	0.6347±0.0081	0.6434±0.0104	0.4807±0.0038	0.6448±0.0131	0.6356±0.0150	0.6463±0.0056
Society	0.6269±0.0076	**0.6540±0.0153**	0.6283±0.0117	0.6422±0.0064	0.6071±0.0145	0.6439±0.0097	0.6364±0.0132	0.6381±0.0178
Entertainment	0.6790±0.0049	**0.7027±0.0065**	0.6864±0.0103	0.6934±0.0068	0.5459±0.0144	0.6987±0.0060	0.6889±0.0129	0.6954±0.0076
Medical	0.9026±0.0114	0.9085±0.0119	0.8980±0.0190	0.8996±0.0185	0.8041±0.0251	0.8675±0.0168	**0.9142±0.0190**	0.8498±0.0148
Cal500	0.4984±0.0120	0.4964±0.0086	**0.5059±0.0043**	0.5016±0.0040	0.4938±0.0073	0.4975±0.0086	0.5026±0.0168	0.4983±0.0112
Social	0.7568±0.0071	**0.7931±0.0043**	0.7612±0.0108	0.7786±0.0104	0.7594±0.0107	0.7868±0.0078	0.7720±0.0167	0.7778±0.0086

Table 4 The results (mean±std) of all 10 datasets in One Error (OE ↓)

Datasets	BDLS	BDLS -SVM	LLSF	LSML	KNN	LIFT	LSF-CI	JLCLS
Arts	0.4696±0.0198	**0.4414±0.0155**	0.4636±0.0081	0.4466±0.0149	0.6090±0.0133	0.4562±0.0056	0.4536±0.0249	0.4576±0.0173
Computers	0.3992±0.0109	**0.3388±0.0151**	0.3570±0.0156	0.3458±0.0140	0.4292±0.0175	0.3510±0.0062	0.3486±0.0103	0.3628±0.0065
emotion	0.2495±0.0164	0.2463±0.0286	0.2479±0.0107	0.2697±0.0362	0.2900±0.0209	0.2546±0.0257	**0.2431±0.0745**	0.3288±0.0164
Business	0.1266±0.0105	**0.1032±0.0065**	0.1232±0.0050	0.1106±0.0057	0.1164±0.0125	0.1056±0.0046	0.1198±0.0156	0.1202±0.0059
Recreation	0.4640±0.0116	**0.4348±0.0157**	0.4458±0.0105	0.4386±0.0119	0.6676±0.0073±	0.4460±0.0177	0.4512±0.0202	0.4508±0.0082
Society	0.4246±0.0100	0.3808±0.0206	0.3952±0.0147	0.3844±0.0111	0.4412±0.0204	0.3870±0.0149	**0.3590±0.0192**	0.4070±0.0255
Entertainment	0.4192±0.0029	**0.3810±0.0059**	0.3922±0.0121	0.3918±0.0083	0.6320±0.0216	0.3860±0.0101	0.3922±0.0169	0.3904±0.0082
Medical	0.1370±0.0267	**0.1166±0.0104**	0.1400±0.0277	0.1391±0.0319	0.2516±0.0218	0.1656±0.0203	0.1176±0.0294	0.2025±0.0259
Cal500	0.1254±0.0167	0.1153±0.0256	0.1175±0.0274	0.1176±0.0133	0.1180±0.0423	0.1197±0.0265	**0.1148±0.0282**	0.1154±0.0176
Social	0.3202±0.0126	**0.2618±0.0085**	0.2724±0.0139	0.2732±0.0081	0.3122±0.01369	0.2682±0.0060	0.2776±0.0234	0.2846±0.0139

Table 5 The results (mean±std) of all 10 datasets in Ranking Loss (RL ↓)

Datasets	BDLS	BDLS -SVM	LLSF	LSML	KNN	LIFT	LSF-CI	JLCLS
Arts	0.1148±0.0020	**0.1089±0.0044**	0.1646±0.0081	0.1325±0.0060	0.1493±0.0054	0.1146±0.0035	0.1421±0.0081	0.1205±0.0034
Computers	0.0771±0.0041	**0.0670±0.0025**	0.1153±0.0046	0.0889±0.0017	0.0866±0.0051	0.0685±0.0037	0.1004±0.0113	0.0792±0.0044
emotion	0.1518±0.0141	**0.1447±0.0103**	0.1572±0.0091	0.1591±0.0093	0.1617±0.0144	0.1462±0.0127	0.1562±0.0294	0.1947±0.0091
Business	0.0404±0.0032	**0.0339±0.0041**	0.0496±0.0037	0.0406±0.0023	0.0376±0.0045	0.0341±0.0032	0.0465±0.0052	0.0369±0.0023
Recreation	0.1237±0.0052	**0.1209±0.0035**	0.1610±0.0066	0.1428±0.0062	0.1835±0.0022	0.1211±0.0057	0.1479±0.0087	0.1277±0.0051
Society	0.1228±0.0051	**0.1173±0.0084**	0.1614±0.0110	0.1449±0.0032	0.1338±0.0052	0.1203±0.0066	0.1525±0.0092	0.1260±0.0074
Entertainment	0.0944±0.0035	**0.0878±0.0024**	0.1281±0.0102	0.1093±0.0058	0.1266±0.0050	0.0891±0.0032	0.1202±0.0103	0.0967±0.0048
Medical	**0.0155±0.0031**	0.0206±0.0047	0.0227±0.0066	0.0159±0.0041	0.0442±0.0118	0.0290±0.0081	0.0174±0.0067	0.0221±0.0070
Cal500	0.1855±0.0051	0.1823±0.0042	**0.1782±0.0041**	0.1790±0.0035	0.1826±0.0041	0.1825±0.0038	0.1803±0.0079	0.1804±0.0057
Social	0.0504±0.0036	**0.0464±0.0066**	0.0871±0.0035	0.0660±0.0073	0.0524±0.0042	0.0479±0.0022	0.0763±0.0088	0.0553±0.0019

Table 6 The results (mean±std) of all 10 datasets in Coverage (CV ↓)

Datasets	BDLS	BDLS -SVM	LLSF	LSML	KNN	LIFT	LSF-CI	JLCLS
Arts	0.1790±0.0048	**0.1675±0.0050**	0.2379±0.0081	0.2044±0.0121	0.2069±0.0049	0.1747±0.0064	0.2165±0.0104	0.1869±0.0059
Computers	0.1173±0.0054	**0.1037±0.0044**	0.1582±0.0054	0.1312±0.0041	0.1256±0.0074	0.1056±0.0051	0.1449±0.0089	0.1182±0.0070
emotion	0.2923±0.0096	0.2849±0.0154	0.2987±0.014	0.2965±0.0105	0.2954±0.0178	**0.2833±0.0149**	0.2968±0.0271	0.3311±0.0108
Business	0.0783±0.0022	**0.0701±0.0059**	0.0903±0.0056	0.0819±0.0035	0.0724±0.0072	0.0706±0.0028	0.0920±0.0078	0.0756±0.0046
Recreation	0.1679±0.0069	**0.1638±0.0065**	0.2111±0.0061	0.1927±0.0092	0.2232±0.0030	0.1641±0.0091	0.1979±0.0119	0.1742±0.0076
Society	0.1941±0.0051	**0.1835±0.0072**	0.2471±0.0130	0.2307±0.0039	0.1999±0.0058	0.1892±0.0083	0.2406±0.0135	0.2018±0.0255
Entertainment	0.1334±0.0042	**0.1244±0.0043**	0.1724±0.0118	0.1523±0.0073	0.1606±0.0075	0.1256±0.0030	0.1656±0.0129	0.1369±0.0049
Medical	**0.0250±0.0045**	0.0343±0.0063	0.0333±0.0058	0.0255±0.0049	0.0627±0.0148	0.0418±0.0108	0.0269±0.0088	0.0336±0.0098
Cal500	0.7613±0.0165	0.7466±0.0148	0.7431±0.0118	**0.7429±0.0093**	0.7473±0.0125	0.7501±0.0060	0.7448±0.0129	0.7439±0.0137
Social	0.0726±0.0032	**0.0669±0.0021**	0.1212±0.0072	0.0967±0.0101	0.0729±0.0050	0.0698±0.0056	0.1094±0.0103	0.0814±0.0018

Table 7 The results (mean±std) of all 10 datasets in AUC (AUC ↑)

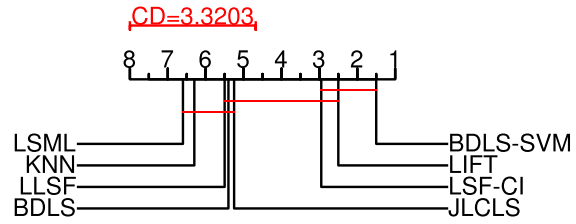
Datasets	BDLS	BDLS -SVM	LLSF	LSML	KNN	LIFT	LSF-CI	JLCLS
Arts	0.8461±0.0032	**0.8582±0.0037**	0.7905±0.0081	0.8259±0.0078	0.8216±0.0043	0.8531±0.0053	0.8156±0.0071	0.8398± 0.0073
Computers	0.8927±0.0035	**0.9053±0.0047**	0.8596±0.0076	0.8825±0.0022	0.8827±0.0065	0.9042±0.0040	0.8713±0.0101	0.8926±0.0071
emotion	0.8186±0.0094	**0.8257±0.0082**	0.8182±0.0049	0.8118±0.0077	0.8093±0.0171	0.8249±0.0096	0.8141±0.0206	0.7766±0.9405
Business	0.9380±0.0045	0.9461 ±0.0060	0.9236±0.0057	0.9351±0.001	0.9417±0.0067	**0.9461±0.0032**	0.9284±0.0068	0.9405±0.0032
Recreation	0.8363±0.0046	0.8428±0.0067	0.7983±0.0061	0.8175±0.0072	0.7840±0.0016	**0.8432±0.0068**	0.8131±0.0118	0.8322±0.0070
Society	0.8340±0.0060	**0.8439±0.0058**	0.7898±0.0106	0.8047±0.0041	0.8306±0.0053	0.8391±0.0065	0.7967±0.0136	0.8284±0.0054
Entertainment	0.8744±0.0032	0.8834±0.0037	0.8596±0.0076	0.8825±0.0022	0.8827±0.0065	**0.9042±0.0040**	0.8466±0.0105	0.8719±0.0047
Medical	**0.9803±0.0026**	0.9725±0.0053	0.9732±0.0044	0.9797±0.0043	0.9474±0.0125	0.9632±0.0098	0.9788±0.0073	0.9723±0.0078
Cal500	0.8115±0.0048	0.8143±0.0041	**0.8189±0.0044**	0.8176±0.0039	0.8140±0.0042	0.8140±0.0043	0.8174±0.0063	0.8159±0.0063
Social	0.9258±0.0038	**0.9321±0.0034**	0.8780±0.0057	0.9030±0.0088	0.9256±0.0058	0.9299±0.0069	0.8928±0.0101	0.9182±0.0018

Table 8 Summary of the Friedman Statistics $F_F(k = 8, N = 10)$ and the critical value in each evaluation metric (k: Comparing Algorithms; N: Data sets)

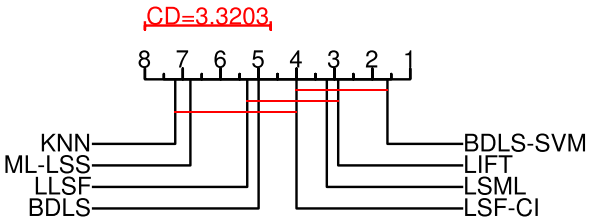
Metric	F_F	Critical value($\rho = 0.05$)
Hamming Loss	13.8054	
Average Precision	9.8811	
One Error	14.1051	2.1588
Ranking Loss	7.1401	
Coverage	6.9091	
AUC	5.7772	

performance of BDLS, BDLS-SVM with other algorithms on ten datasets. When the difference between the average ranking of the two algorithms on all data sets is less than or equal to the critical difference $CD = \sqrt{k(k+1)/6N}$, the two algorithms are assumed to have no significant difference. The CD diagrams on the six evaluation indicators are shown in Fig. 4. And for each subgraph of Fig. 4, the interconnected algorithms are considered to be the ones with less significant differences. For Nemenyi Test, $q_\alpha = 2.850$ at the significance level of 0.05, the critical difference is $2.3845(k = 6, N = 10)$. Based on these experimental results, the following conclusions are drawn,

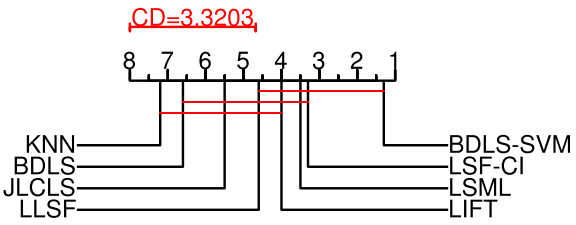
- The BDLS algorithm is superior to LLSF, ML-KNN, LSML, JLCLS and LSF-CI in RL, CV, and AUC. It indicates the effectiveness of the reconstruction loss regularization term used to compensate for the forward projection information loss and the importance of label causality for mining the essential link between labels.
- On the ten datasets, BDLS-SVM algorithm shows the best performance, followed by the LIFT algorithm. Comparing BDLS-SVM with LIFT reveals that BDLS-SVM performs better than LIFT for HL, AP, OE, RL, and CV on eight datasets except for Cal500 and Emotion. It fully illustrates the effectiveness of our proposed label-specific feature learning method.
- For the BDLS algorithm, the performance is unsatisfactory on the dataset Cal500. The intuitive analysis is that the Cal500 dataset causes the BDSL reverse reconstruction to be a mapping compression. Adding the reverse reconstruction loss regularization term increases the loss of information, so BDLS exhibits suboptimal results. However, compared to other methods, the BDLS algorithm shows optimal RL, CV, and AUC results on the Medical dataset. The analysis of the Medical dataset structure reveals that d is about 32.5 times larger than l ($d \gg l$), and the addition of reverse reconstruction loss regularization term can effectively reduce the loss in feature dimensionality reduction and improve the performance of the algorithm.



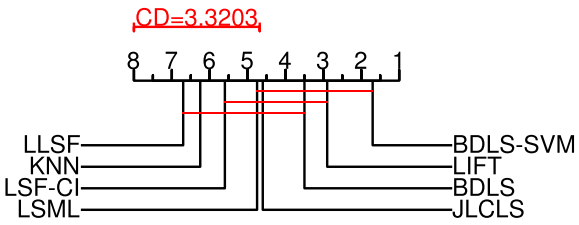
(a) HL↓



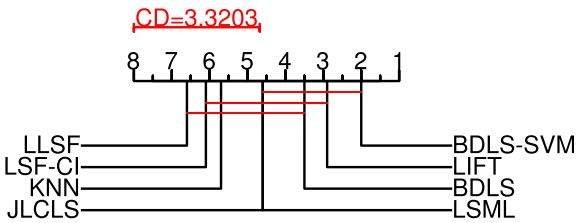
(b) AP↑



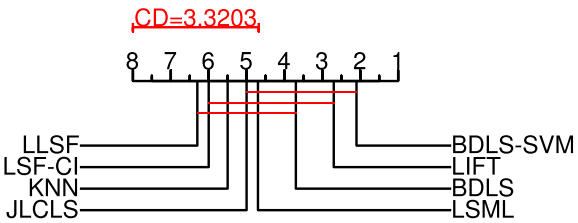
(c) OE↓



(d) RL↓



(e) CV↓

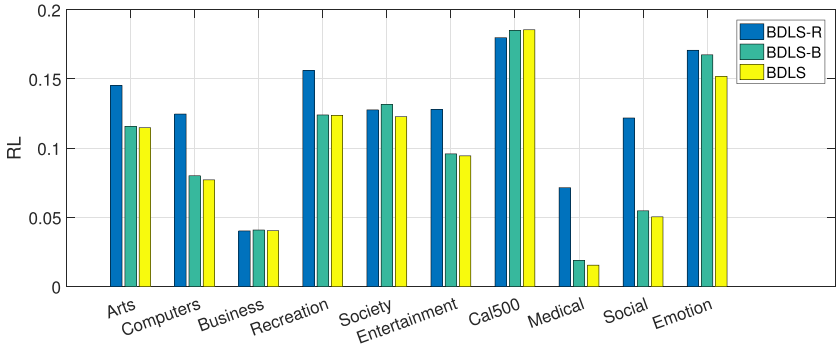


(f) AUC↑

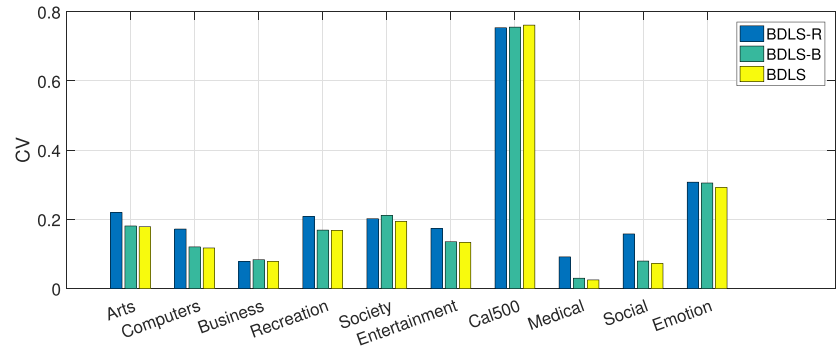
Fig. 4 Comparison of BDLS or BDLS-SVM against other comparing algorithms with the Nemenyi test

4.5 Component analysis

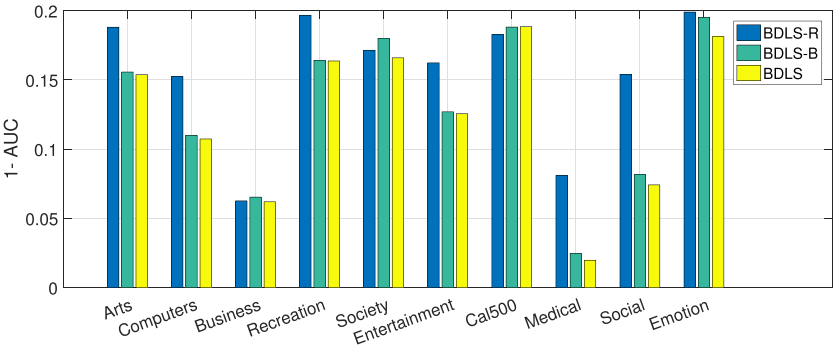
To further validate the effectiveness of reconstructing loss regularization term and label causality in BDLS, we conducted comparative experiments using BDLS-R and BDLS-B with BDLS on ten datasets. BDLS-R and BDLS-B ignore the reconstruction loss regularization term and label causality, respectively, based on BDLS. Figure 5 indicates



(a) RL↓



(b) CV↓

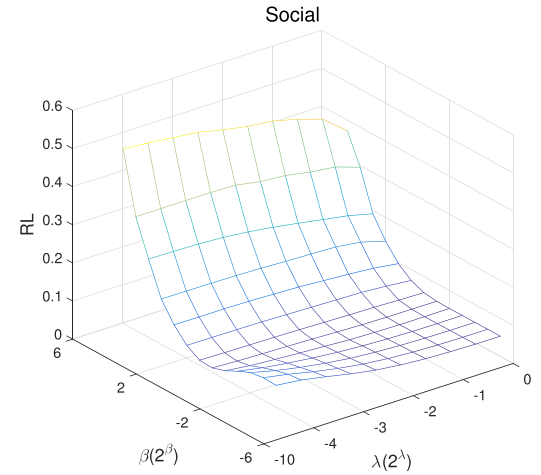


(c) 1-AUC↓

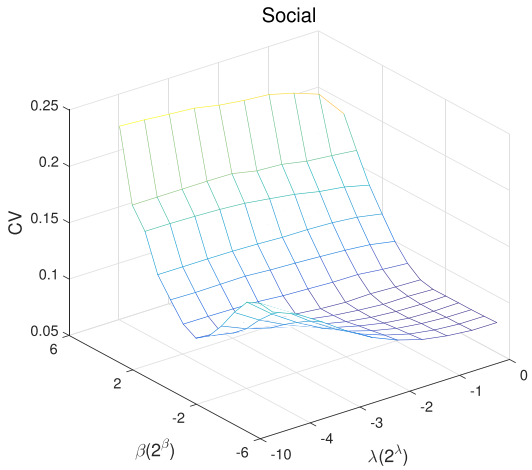
Fig. 5 Performance comparison of BDLS with BDLS-B and BDLS-R on all datasets

the histograms of RL, CV, and 1-AUC for BDLS-R, BDLS-B, and BDLS on ten data sets. From Fig. 5, we can draw the following conclusions,

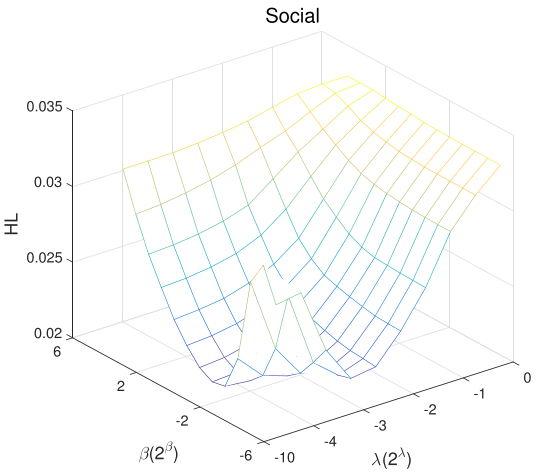
- BDLS algorithms outperform both variants of the algorithm on most datasets. The primary reason is that BDLS improves the algorithm performance by utilizing both parts of information.
- On the dataset Cal500, BDLS-R has better results than BDLS-B and BDLS. It is found that the number of instances is much smaller than the number of tags by analyzing the data set structure of Cal500. The mapping of instance space to label space, in other words, is not a mapping compression but an expansion. In this case, reverse mapping is a type of mapping compression. So when considering the reconstruction loss, increasing the loss of label data may make BDLS and BDLS-B algorithms exhibit suboptimal results.
- Comparing BDLS-B and BDLS-R, BDLS-B is better than BDLS-R in many cases, indicating that reconstruction loss regularity terms play a more critical role



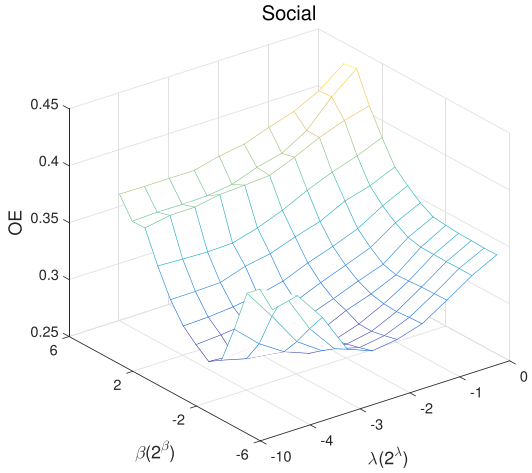
(a) RL↓



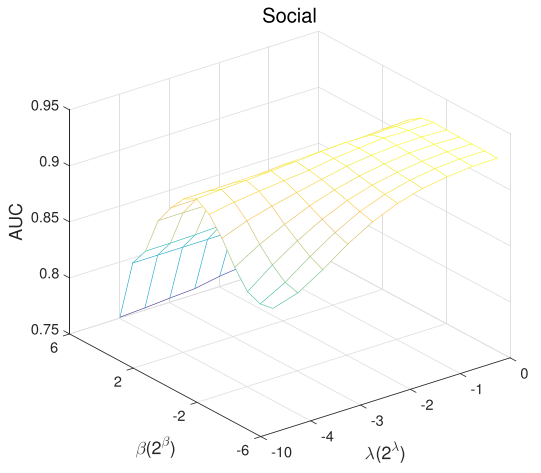
(b) CV↓



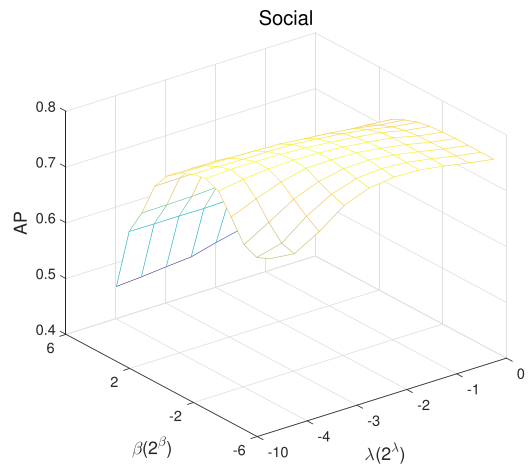
(c) HL↓



(d) OE↓



(e) AUC↑



(f) AP↑

Fig. 6 Parameter sensitivity analysis over the reconstruction loss factor λ and the sparsity factor β in the Social datasets. The value of α is 2^{-1}, $\beta \in \{2^{-6}, 2^{-5}, \cdots, 2^{5}, 2^{6}\}$, and $\lambda \in \{2^{-10}, 2^{-9}, \cdots, 2^{-2}, 2^{-1}\}$

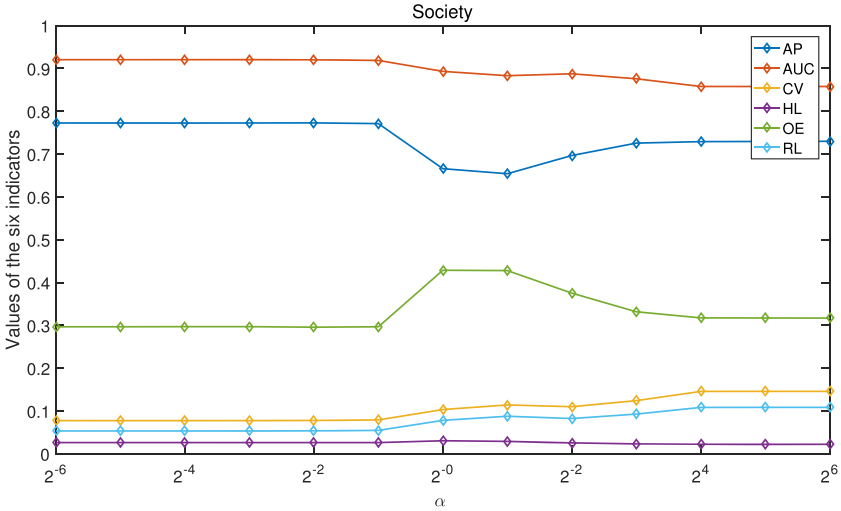


Fig. 7 Parameter sensitivity analysis over the label causality factor α in the datasets social, where $\beta = 2^{-1}$, $\lambda = 2^{-3}$ and $\alpha \in \{2^{-6}, 2^{-5}, \cdots, 2^{5}, 2^{6}\}$

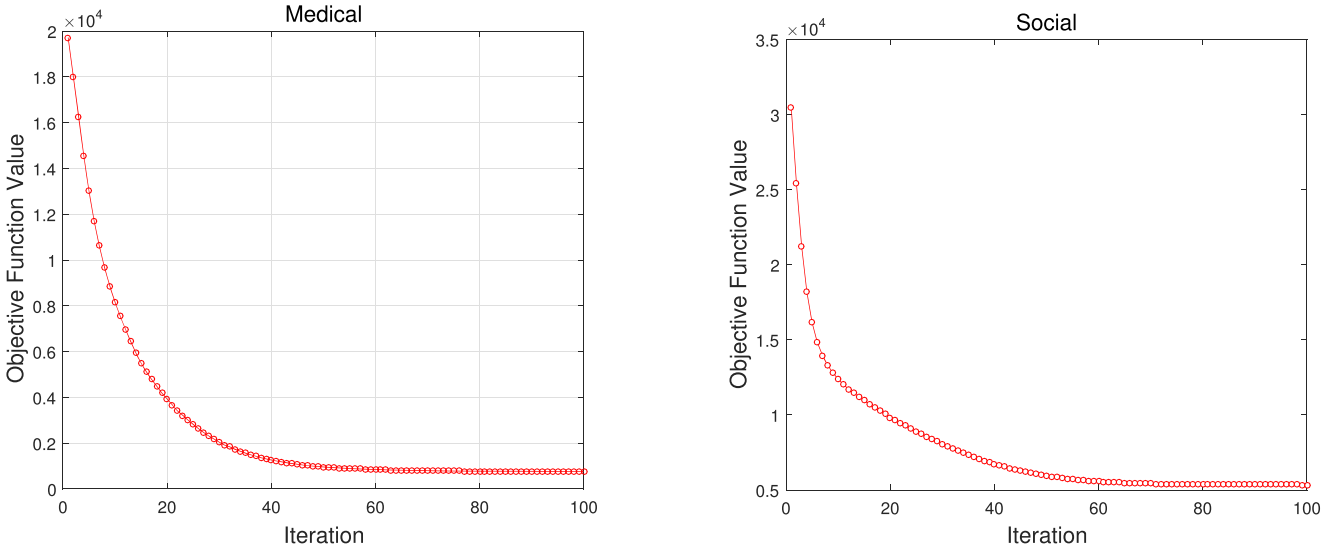
than label causality in multi-label learning of BDLS. BDLS will achieve better performance if better algorithms are used to model the correlations between the labels.

4.6 Parameter sensitivity analysis

In this paper, experiments are conducted on the Social dataset to investigate the sensitivity of BDLS to α, β, and λ, where α controls label causality, β rules regulate $\boldsymbol{W}$ sparsity, and λ balances the reconstruction loss. Figure 6 and Fig. 7 show the variation curves of the six evaluation indicators with α, β, and λ. From Fig. 6, it can be seen that BDLS algorithm shows optimal results when β is around 2^{-1}. Specifically, when $\beta \leq 2^{-1}$, insufficient sparsity of W leads to the inability to select label-specific features, and when $\beta \geq 2^{-1}$, extreme sparsity of W leads to the loss of feature information. Similarly, When $\lambda \leq 2^{-4}$, the performance of BDLS algorithm decreases because of the slight reconstruction loss regularization term used to compensate for the information loss in the forward mapping. However, when $\lambda \leq 2^{-4}$, the performance of BDLS algorithm is significantly improved, further illustrating the effectiveness of the reconstruction loss regularization terms. From Fig. 7, we can observe that the performance of BDLS algorithm is best when α is near 2^{-1}. In addition, if α is oversize, some weakly related category labels are considered to be strongly correlated, which leads to suboptimal multi-label classification results.

4.7 Run time and convergence analysis

To analyze the competitiveness of the proposed methods in terms of running time, Fig. 8 shows the experimental results



(a) Medical (b) Social

Fig. 8 Convergence trend analysis

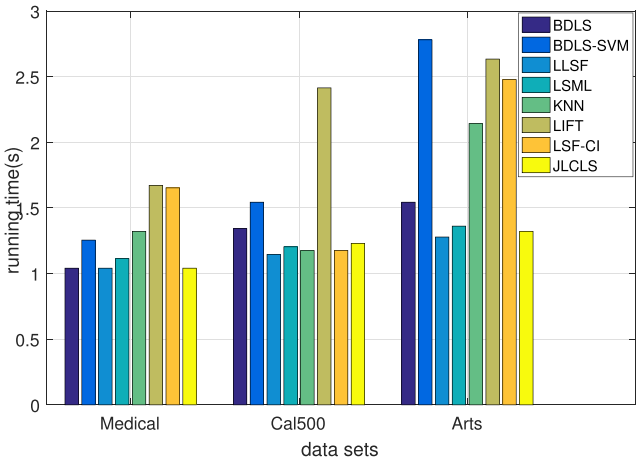


Fig. 9 Comparison of the running time of 8 algorithms on the Medical, Cal500, and Arts datastets, assuming the running time is t, the Y-axis represents log(t)

of BDLS and BDLS-SVM compared with other algorithms running time on three datasets. The experimental method is the same as that in Subsection 4.4.2, and the experiments are operated in Windows 10, Intel(R) Core(TM) i7-7700K, 4.20GHz CPU and Matlab 2016a. As can be seen from Fig. 8, BDLS is comparable to LLSF, LSML, and JLCLS in terms of running time in most cases, while the remaining four algorithms take relatively longer time. Moreover, the BDLS-SVM algorithm and LIFT consume significantly more time when the dataset size is larger. This is due to the inclusion of the more time-consuming binary classifier SVM module in both algorithms, which causes a sharp increase in the running time cost. These comparisons confirm the validity of the proposed model.

Figure 9 shows the convergence curves of BDLS algorithm on the Medical and Social datasets. In the experiment, the maximum number of iterations is 100. As can be seen from Fig. 9, BDLS converges after 80 iterations on both datasets. After investigations, similar convergence trends were observed on all other datasets.

5 Conclusion

In this paper, we propose a multi-label learning algorithm based on bi-directional mapping and label causality. Bi-directional mapping is used to explore label-specific features, and label causality mines the intrinsic relationships between labels. Firstly, we establish the basic framework of a multi-label learning model for learning specific features of labels by projection loss and reconstruction loss. Then causal inference between labels is introduced to explore the interrelationships between labels. Finally, the obtained model coefficients are used to predict labels for unknown instances. The experimental results show that the proposed reconstruction loss and label causality have great generalization ability over multiple datasets. The algorithm in this paper has a better performance compared with several state-of-the-art algorithms. However, the results of our proposed algorithm are unsatisfactory when dealing with multi-label datasets where the number of labels is larger than the number of instances, and this is the part that we will optimize and study subsequently. Next, we will conduct experiments on more classed datasets to investigate how to solve multi-label classification problems where the label dimension is larger than the instance dimension. In addition, we will also explore the higher-order causal relationships between labels.

Acknowledgements This work is supported by the National Natural Science Foundation of China (No. 62071001), the Anhui Natural Science Foundation of China (Nos. 2008085MF192 and 2008085MF183), the Key Science Project of Anhui Education Department of China (Nos. KJ2018A0012, KJ2019A0023, and KJ2019A0022), and the CERNET Innovation Project of China (Nos. NGII20180612, NGII20180312, and NGII20180624).

References

1. Tsoumakas G, Katakis I (2007) Multi-label classification: An overview. Int J Data Warehous Min (IJDWM) 3(3):1–13
2. Zhang M-L, Zhou Z-H (2013) A review on multi-label learning algorithms. IEEE Trans Knowl Data Eng 26(8):1819–1837
3. Ueda N, Saito K (2003) Parametric mixture models for multi-labeled text. In: Advances in neural information processing systems, pp 737–744
4. Schapire RE, Boostexter YS (2000) A boosting-based system for text categorization. Mach Learn 39(2):135–168
5. Qi G-J, Hua X-S, Rui Y, Tang J, Mei T, Zhang H-J (2007) Correlative multi-label video annotation. In: Proceedings of the 15th ACM international conference on Multimedia, pp 17–26
6. Barutcuoglu Z, Schapire RE, Troyanskaya OG (2006) Hierarchical multi-label prediction of gene function. Bioinformatics 22(7):830–836
7. Trohidis K, Tsoumakas G, Kalliris G, Vlahavas IP (2008) Multi-label classification of music into emotions. ISMIR 8:325–330
8. Wu B, Zhong E, Horner A, Yang Q (2014) Music emotion recognition by multi-label multi-layer multi-instance multi-view learning. In: Proceedings of the 22nd ACM international conference on Multimedia, pp 117–126
9. Boutell MR, Luo J, Shen X, Brown CM (2004) Learning multi-label scene classification. Pattern Recogn 37(9):1757–1771
10. Zhang M-L, Zhou Z-H (2007) Ml-knn: A lazy learning approach to multi-label learning. Pattern Recogn 40(7):2038–2048
11. Gibaja E, Ventura S (2015) A tutorial on multilabel learning. ACM Comput Surv (CSUR) 47(3):1–38
12. Han H, Huang M, Yu Z, Yang X, Feng W (2019) Multi-label learning with label specific features using correlation information. IEEE Access 7:11474–11484
13. Zhang M-L, Wu L (2014) Lift: Multi-label learning with label-specific features. IEEE Trans Pattern Anal Mach Intell 37(1):107–120

14. Ma Z, Nie F, Yi Y, Uijlings JRR, Sebe N (2012) Web image annotation via subspace-sparsity collaborated feature selection. IEEE Trans Multimed 14(4):1021–1030
15. Jian L, Li J, Shu K, Liu H (2016) Multi-label informed feature selection. IJCAI 16:1627–33
16. Chang X, Nie F, Yi Y, Huang H (2014) A convex formulation for semi-supervised multi-label feature selection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 28
17. Li J, Zhang C, Zhu P, Wu B, Chen L, Hu Q (2020) Spl-mll: Selecting predictable landmarks for multi-label learning. In: European Conference on Computer Vision. Springer, pp 783–799
18. Huang J, Li G, Huang Q, Wu X (2017) Joint feature selection and classification for multilabel learning. IEEE Trans Cybern 48(3):876–889
19. Huang J, Li G, Huang Q, Wu X (2016) Learning label-specific features and class-dependent labels for multi-label classification. IEEE Trans Knowl Data Eng 28(12):3309–3323
20. Pearl J, Mackenzie D (2018) The book of why: the new science of cause and effect. Basic books
21. Gibaja E, Ventura S (2014) Multi-label learning: a review of the state of the art and ongoing research. Wiley Interdiscip Rev Data Min Knowl Discov 4(6):411–444
22. Tsoumakas G, Vlahavas I (2007) Random k-labelsets: An ensemble method for multilabel classification. In: European conference on machine learning. Springer, pp 406–417
23. Read J, Pfahringer B, Holmes G (2008) Multi-label classification using ensembles of pruned sets. In: 2008 eighth IEEE international conference on data mining. IEEE, pp 995–1000
24. Hüllermeier E, Fürnkranz J, Cheng W, Brinker K (2008) Label ranking by learning pairwise preferences. Artif Intell 172(16-17):1897–1916
25. Fürnkranz J, Hüllermeier E, Loza mencía E, Brinker K (2008) Multilabel classification via calibrated label ranking. Mach Learn 73(2):133–153
26. Zhang M-L, Zhou Z-H (2007) Multi-label learning by instance differentiation. AAAI 7:669–674
27. Elisseeff A, Weston J et al (2001) A kernel method for multi-labelled classification. NIPS 14:681–687
28. Clare A, King RD (2001) Knowledge discovery in multi-label phenotype data. In: European conference on principles of data mining and knowledge discovery. Springer, pp 42–53
29. Zhang M-L, Zhou Z-H (2006) Multilabel neural networks with applications to functional genomics and text categorization. IEEE Trans Knowl Data Eng 18(10):1338–1351
30. Cheng W, Hüllermeier E (2009) Combining instance-based learning and logistic regression for multilabel classification. Mach Learn 76(2-3):211–225
31. Tsoumakas G, Katakis T, Vlahavas T (2009) Mining multi-label data. In: Data mining and knowledge discovery handbook. Springer, pp 667–685
32. Zhang M-L, Zhang K (2010) Multi-label learning by exploiting label dependency. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 999–1008
33. Liu H, Li X, Zhang S (2016) Learning instance correlation functions for multilabel classification. IEEE Trans Cybern 47(2):499–510
34. Gong C, Tao D, Yang J, Liu W (2016) Teaching-to-learn and learning-to-teach for multi-label propagation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 30
35. Zhu S, Ji X, Xu W, Gong Y (2005) Multi-labelled classification using maximum entropy method. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp 274–281
36. Li Y-K, Zhang M-L, Geng X (2015) Leveraging implicit relative labeling-importance information for effective multi-label learning. In: IEEE International Conference on Data Mining. IEEE, pp 251–260
37. Godbole S, Sarawagi S (2004) Discriminative methods for multi-labeled classification. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, pp 22–30
38. Yan R, Tesic J, Smith JR (2007) Model-shared subspace boosting for multi-label classification. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 834–843
39. Read J, Pfahringer B, Holmes G, Frank E (2011) Classifier chains for multi-label classification. Mach Learn 85(3):333
40. Charte F, Rivera AJ, Del Jesus MJ, Herrera F (2014) Li-mlc: a label inference methodology for addressing high dimensionality in the label space for multilabel classification. IEEE Trans Neural Netw Learn Syst 25(10):1842–1854
41. Zhang J-J, Fang M, Li X (2015) Multi-label learning with discriminative features for each label. Neurocomputing 154:305–316
42. Guo Y, Chung F, Li G, Wang J, Gee JC (2019) Leveraging label-specific discriminant mapping features for multi-label learning. ACM Trans Knowl Discov Data (TKDD) 13(2):1–23
43. Sun L, Kudo M, Kimura K (2016) Multi-label classification with meta-label-specific features. In: 2016 23rd International Conference on Pattern Recognition (ICPR). IEEE, pp 1612–1617
44. Huang J, Li G, Huang Q, Wu X (2015) Learning label specific features for multi-label classification. In: 2015 IEEE International Conference on Data Mining. IEEE, pp 181–190
45. He Z-F, Yang M (2019) Sparse and low-rank representation for multi-label classification. Appl Intell 49(5):1708–1723
46. 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. Inf Sci 492:124–146
47. Cheng Y, Zhao D, Wang Y, Pei G (2019) Multi-label learning with kernel extreme learning machine autoencoder. Knowl-Based Syst 178:1–10
48. Ling Z, Yu K, Zhang Y, Liu L, Li J (2021) Causal learner: A toolbox for causal structure and markov blanket learning. arXiv:2103.06544
49. Margaritis D, Thrun S (1999) Bayesian network induction via local neighborhoods. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE
50. Yang S, Wang H, Yu K, Cao F, Wu X (2019) Towards efficient local causal structure learning. arXiv:1910.01288
51. Beck A, Teboulle M (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J Imaging Sci 2(1):183–202
52. Wang Y, Zheng W, Cheng Y, Zhao D (2020) Joint label completion and label-specific features for multi-label learning algorithm. Soft Comput 24(9):6553–6569
53. Bucak SS, Jin R, Jain AK (2011) Multi-label learning with incomplete class assignments. In: CVPR 2011. IEEE, pp 2801–2808
54. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

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