International Journal of Machine Learning and Cybernetics (2023) 14:1161–1177
https://doi.org/10.1007/s13042-022-01692-7

ORIGINAL ARTICLE

Learning label-specific features via neural network for multi-label classification

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

Received: 1 December 2021 / Accepted: 13 October 2022 / Published online: 11 November 2022

Abstract
In multi-label learning, learning specific features for each label is an effective strategy, and most of the existing multi-label classification methods based on label-specific features commonly use the original feature space to learn specific features for each label directly. Due to the problem of dimensionality disaster in the feature space, it may not be the optimal strategy to directly generate the specific feature of the label in the original feature space. Therefore, this paper proposes a multi-label learning framework that joins neural networks and label-specific features. First, the neural network projects the original feature space to a low-dimensional mapping space to learn potential low-dimensional feature space representations, and this nonlinear feature mapping can mine the potential feature information inside the complex feature space. Then, in the low-dimensional mapping space, specific features of the labels are learned using empirical minimization loss. Finally, a unified multi-label classification model is constructed by considering label correlation and instance similarity issues. Extensive experiments are conducted on 12 different multi-label data sets and demonstrate the better generalizability of our proposed approaches.

Keywords Multi-label learning · Label-specific features · Neural network · Label correlation

1 Introduction

In traditional supervised learning, there is a one-to-one correspondence between data samples and category labels, that is, a single data sample is only associated with one category label. However, in the reality, objects tend to have multiple semantics. For example, a picture can be annotated as "blue sky", "white clouds" and "lake" simultaneously, and there may be a strong correlation among labels. Nowadays, multi-label learning has become one of the important research hotspots in data mining and machine learning, and its main task is to assign the corresponding category labels to the objects to be classified. Researchers have enthusiastically proposed many mature multi-label classification algorithms, which have been widely applied in various research areas. For example, text classification [1], image annotation [2, 3], bioinformatics [4, 5] etc.

Multi-label classification algorithms are often classified into the following two categories [6]: problem transformation methods and algorithm adaptive methods. Specifically, the problem transformation approach transforms a multi-label learning problem into one or more traditional single-label learning problems. Its representative algorithm, such as BR [7], the core idea is to decompose the multi-label learning problem into several unrelated single-label learning problems, and then use mature and advanced methods to take effective solutions to these learning subtasks. Algorithm adaptive methods improve the traditional supervised learning algorithms to be applicable to the prediction of multi-label data. The representative algorithm is ML-KNN [8], which classifies the predicted samples based on the Maximum A Posteriori Probability rule using the label information of the sample's neighboring locations. However, all of them ignore the correlation between labels, which reduces the learning effect of multi-label classification models. Therefore a large number of correlation-based methods have been proposed one after another. Based on the different label correlation strategies, the multi-label classification

✉ Dong Sun
sundong@ahu.edu.cn

Ling Jia
lingjiash@163.com

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

algorithms can be classified into first-order strategy [7, 8], second-order strategy [9, 10], and higher-order strategy [11, 12] respectively.

Similar to single-label classification, the feature space of multi-label classification is usually high-dimensional, which easily causes the problem of dimensional catastrophe. Recently, many dimensionality reduction methods have been applied to multi-label classification tasks [13, 14]. These methods are mostly based on the fact that each label has the same feature space. In real life, however, each label may be determined by its unique subset of features. For example, in image classification, color-based features are most beneficial to distinguishing between blue sky and white clouds in images. While texture-based features are most helpful to distinguish between desert and hills. To solve the above problems, many new algorithms have been proposed to select a set of feature subsets with good distinguishing characteristics, and effectively eliminate the redundant features in the correlation features [15–17], so as to achieve the reduction of feature dimensionality and improve the accuracy of the classification model, while the specific feature subsets extracted are more beneficial to the classification effect of the model.

With the rapid development of deep learning, neural network-based modeling methods have greatly promoted the progress of multi-label classification research. The neural network is formed by connecting many neurons with adjustable connection weights and has good self-organization and self-learning capabilities. Zhang et al. [18] developed a backpropagation algorithm for multi-label learning(BP-MLL), which is an adaptation of traditional multilayer feedforward neural networks for multi-label data. The core idea is to capture the features of multi-label learning by minimizing the global error function. Marilyn et al. [19] proposed a bidirectional neural network structure to learn the correlation among labels. Other CNN and RNN-based neural network algorithms are adapted to solve multi-label prediction problems [20–22].

In summary, existing multi-label classification methods have achieved good achievements in capturing information from original data and in establishing correlations between labels. However, the following three challenges exist:

- Most of the previous research methods mainly used the same feature data set to represent each category label, which not only increased the complexity of calculation but also was not conducive to distinguishing and expressing the attribute information of each label.
- Existing multi-label learning algorithms are trained and predicted on multi-label datasets in the original feature space. With the explosive growth of feature dimensions, it will become very challenging to capture the internal laws of the instance feature space. Such learning may lead to an over dimensionality of the feature space that is difficult to visualize.
- Although considering the interrelationship among labels can improve the classification accuracy, as yet the intrinsic correlation among different instance samples is often ignored, and mining the correlation information of the instances can facilitate the training effect of the model and achieve the purpose of improving the classification performance.

In order to solve the above-mentioned problems, in this paper, we propose an algorithm to learn label-specific features via neural network for multi-label classification(LLFN). First, we represent the original feature space of the input data by a neural network with a low-dimensional mapping: $X \rightarrow \hat{X}$, and this nonlinear feature mapping can mine the feature information inside the complex feature space, visualize high-dimensional data and maintain the topology of the input space structure. According to this internal feature space information, we then employ the common squared minimization loss function to model the basic framework for label-specific feature learning. Based on this, we also introduce label correlation and instance similarity to optimize the model. A unified end-to-end multi-label classification framework is finally constructed. The specific model diagram is shown in Fig. 1.

The main contributions of the research in this paper are as follows:

- Different from the traditional multi-label classification method, this paper uses a single hidden layer neural network to learn the latent representation of the feature and extracts the specific feature of the label in the latent feature space.
- This is an end-to-end multi-label classifier with a label-specific feature-based joint learning model.
- Experimental results on 12 widely used datasets show that our proposed method achieves some advantages over the state-of-the-art algorithms.

The rest of the paper is organized as follows. Section 2 provides extensively referenced work, giving an introduction to previous neural network multi-label learning algorithms and multi-label specific feature learning. The LLFN algorithm process is introduced in Sect. 3. Section 4 presents the experimental results and experimental analysis, and finally, the paper is summarized in Sect. 5.

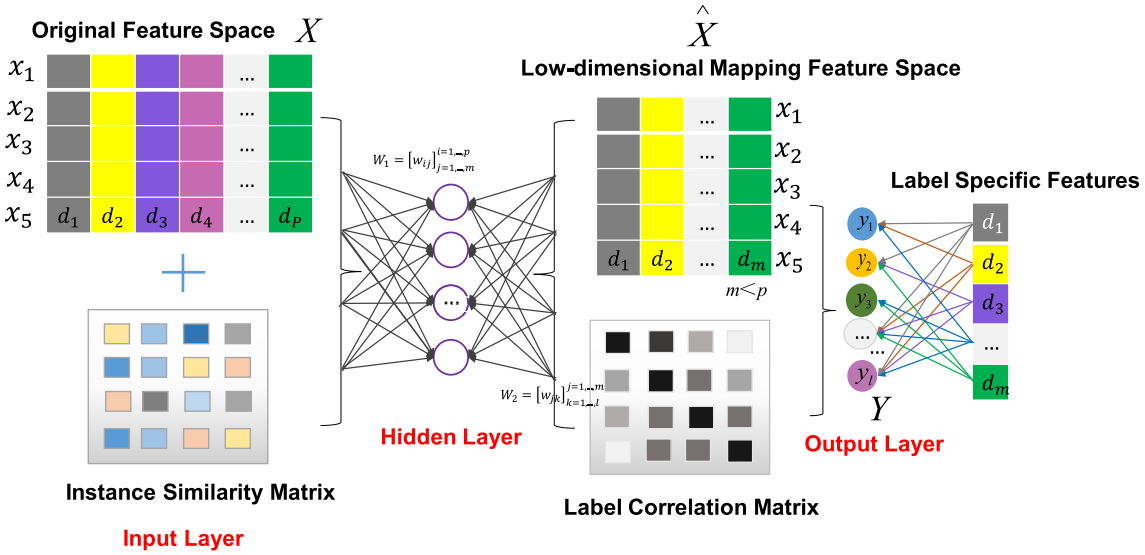


Fig. 1 Model framework of LLFN

2 Related work

2.1 Neural network multi-label learning

Neural networks have a widespread application in multi-label learning, and many algorithms for neural network multi-label learning have been generated in the past decade or so. Zhang et al. [18] were the first to propose the application of neural networks to multi-label classification and achieved good results compared to traditional machine learning methods. This paper used the backpropagation of multi-label learning(BP-MLL) neural network algorithm, which captures the features of multi-label learning by minimizing the inter-label sorting error. However, for large-scale multi-label text classification, the BP-MLL algorithm shows limitations. Nam et al. [20] proposed a single hidden layer of neural network architecture. The model replaces the ranking loss minimization with a cross-entropy error function on basis of BP-MLL. It demonstrates that a simple network configuration makes the model scale better and is more suitable for large-scale text classification tasks. Subsequently, Zhang [23] also proposed an RBF neural network-based multi-label learning algorithm, in which the k-means clustering analysis of the instances is first performed by the first neural network layer, and the center of mass of the clustering group is used for the prototype vector of the basis function; then the error function is minimized to learn the second ML-RBF layer weights. In this way, we can make full use of prototype vector encoding information to optimize the output neuron weights. Lu et al. [24] propose a method that uses a combination of fuzzy logic technique and DNN. The deep fuzzy hashing network (DFHN) automatically generates more effective image features for accurate prediction and classification of image datasets. In addition, autoencoders can automatically learn features of data samples [25, 26], Based on this mind, Chen et al. [27] proposed a kernel limit learning machine based auto-encoder based multi-label learning algorithm, which improves multi-label classification performance by reconstructing the label space information with auto- encoder networks, and improved generalizability of the model.

Moreover, convolutional neural networks(CNN) [21, 28, 29] and recurrent neural networks(RNN) [20, 30, 31] are increasingly used in the field of multi-label learning. Liao et al. [21] proposed a multi-label learning algorithm based on convolutional neural networks and fully initialized connections. It is a sequence-to-sequence multi-label classification model using encoders and decoders. In this, the encoder is used to encode semantic information using neural networks and attention mechanisms. The decoder combines LSTM and initialized fully connected layers to mine the global correlation and local correlation of the labels. Chen et al. [31] proposed a recurrent neural network-based multi-label classification architecture for images, which introduces the LSMT model and reflects the dependencies between labels through a visual attention mechanism. In [22], the authors propose a unified multi-label learning framework that combines the advantages of CNN and RNN for image/label embedding. The semantic label dependencies and image-label interrelationships can be learned. The semantic features are first extracted from the images by the CNN part, and then the label dependencies and the picture-label interrelationships are modeled using the RNN part to better predict the probability of labels.

2.2 Label-specific features learning

In multi-label learning, most of the existing algorithms deal with datasets with the same features, however, this is not the most ideal way, as each label tends to have its own inherent feature properties. LPLC-LA [32] is a learning method based on extracting label-specific features for obtaining local positive and negative label correlations and addresses the label imbalance problem using perceptual weights between labels. The above algorithm considered the feature-to-feature dependency but failed to reasonably and effectively eliminate the redundant features in the feature space. Bidgoli et al. [33] proposed a new multi-objective optimization method to reduce the complexity of the model by reducing the number of features; meanwhile, based on correlation analysis and redundancy analysis, it can effectively eliminate the redundancy in related features, thereby improving the classification performance.

Also for label space, using the correlation among labels to guide feature selection can greatly improve the classification performance [16, 34–36]. Huang et al. [16] argue that the strength of correlation among labels is potentially correlated with the magnitude of similarity among features, based on which label-specific features are learned by a linear regression model. To some extent, the method proves to fully exploit the correlation of the labels, and improve the performance of the multi-label learning algorithm. GLOCAL [36] effectively solves the problem of global labels and missing labels by considering global label correlation and shared local label correlation.

In multi-label learning, besides using the potential relationship between labels to provide additional information for multi-label learning, the samples are also correlated with each other [37, 38], Jie et al. [37] proposed a popular regularization-based multi-task feature selection learning method (MTFS), which considers instance similarity by introducing a popular Laplace-based regularization. Han et al. [38] proposed a multi-label learning algorithm that uses correlation information to learn specific features of labels (LSF-CI). LSF-CI considers that if two instances in feature space have a strong correlation, their corresponding labels will be similar.

In the previous research on multi-label learning methods, neural network algorithms have been widely used, and in recent years, a large number of multi-label classification methods that combine label-specific features, label correlation, and instance similarity have been proposed. However, most algorithms extract label-specific features in the original feature space, which is possibly not the most optimal strategy. Therefore, in this paper, we propose a neural network to map the original feature space into the embedded feature space of labels and then perform label-specific feature extraction in the embedded feature space, and finally, the performance and generalization of the algorithm are improved by introducing label correlation and instance similarity.

3 Proposed approach

3.1 Preliminaries

In multi-label learning, the input feature space is assumed to be represented as $\boldsymbol{X} = \left[x_1, \ldots, x_n\right]^{\mathrm{T}} \in \mathbb{R}^{n\times p}$, and represent the output label matrix space as $\boldsymbol{Y} = \left[y_1, \ldots, y_n\right]^{\mathrm{T}} \in \mathbb{R}^{n\times l}$, and the training dataset with n examples is $\boldsymbol{D} = \left\{(x_i, y_i) \mid 1 \le i \le n\right\}$. Denote the p-dimensional feature vector by $x_i = \left[x_{i1}, \ldots, x_{ip}\right]$, $x_i \in \boldsymbol{X}$, and $y_i = \left[y_{i1}, \ldots, y_{il}\right]$ is a l-dimensional real-valued label vector. If the label y_i is associated with x_i, then each element $y_{ij} = 1$, otherwise $y_{ij} = 0$. The task of MLL involves learning a function $h : \boldsymbol{X} \to 2^{\boldsymbol{Y}}$ from the multi-label set of training that predicts the confidence of each label by the mapping function $h(\cdot)$ for any invisible instance $x \in \boldsymbol{X}$.

3.2 Learning multi-label specific features based on neural networks

As mentioned above, each category label has its own specific features. However, in previous studies, the specific feature of the label is a subspace filtered from the original feature space, and the subspace is relatively sparse compared to the original feature space. As shown in Fig. 1, we propose a potential mapping representation of instance features obtained by a low-dimensional mapping of the input feature space by a neural network. The neural network structure in Fig. 1 includes an input layer $\boldsymbol{X}$, an output layer $\boldsymbol{Y}$, and a hidden layer, where the weight coefficient matrices connected to the hidden layer are $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$, respectively. In this paper, the activation function of the hidden layer is the hyperbolic tangent function $\tanh(\cdot)$. Our model can be initially expressed as

$$\min_{\boldsymbol{W}_1, \boldsymbol{W}_2} \frac{1}{2}\left\|\tanh(\boldsymbol{X}\boldsymbol{W}_1)\boldsymbol{W}_2 - \boldsymbol{Y}\right\|_F^2 + \beta\left\|\boldsymbol{W}_2\right\|_1 + \frac{\gamma}{2}\left\|\boldsymbol{W}_1\right\|_F^2 \quad (1)$$

The first term in Eq. 1 is the squared loss term of the combined neural network. where $\boldsymbol{W}_1 \in \mathbb{R}^{p\times d}$ denotes the weight matrix of neuronal connections between the hidden and input layers, and $\boldsymbol{W}_2 \in \mathbb{R}^{d\times l}$ is the weight matrix between the hidden and output layers. The second term is the l_1-norm regularization term that simulates the sparsity of specific features of the label, and β is the parameter that controls its sparsity. The third term is a regularization term that controls the complexity of the model, and γ is its weight coefficient. Moreover, combining Fig. 1 and Eq. 1, it can be found that

W1 aims at a low-dimensional data representation of the original feature space with a nonlinear mapping through the activation function. $\boldsymbol{W}_2$ aims to learn label-specific features naturally by reserving non-zero feature elements for each label.

3.3 Combining Label Correlations

In multi-label learning, considering label correlation can improve the classification performance of multiple labels. From the work in [16], if two labels have strong correlations, the features contained in one of the labels should be very close to the features possessed by the other label. That is, if the labels y_i and y_j are strongly correlated, the similarity between the coefficient vector w_{2_i} and w_{2_j} will be large, otherwise, the similarity will be small. After introducing label correlation, the objective function is obtained as

$$\min_{\boldsymbol{W}_1,\boldsymbol{W}_2} \frac{1}{2}\left\|\tanh(\boldsymbol{X}\boldsymbol{W}_1)\boldsymbol{W}_2-\boldsymbol{Y}\right\|_F^2+\frac{\alpha}{2}tr(\boldsymbol{W}_2\boldsymbol{R}\boldsymbol{W}_2{}^T) + \beta\left\|\boldsymbol{W}_2\right\|_1+\frac{\gamma}{2}\left\|\boldsymbol{W}_1\right\|_F^2 \tag{2}$$

where $\boldsymbol{R}=1-\boldsymbol{C}$, The element $\boldsymbol{C}_{ij}$ in $\boldsymbol{C}$ represents the similarity between the label y_i and the label y_j. Because the label matrix $\boldsymbol{Y}$ is a binary variable, and the Hamming distance is a good way to measure the similarity of binary variables [39, 40], the Hamming distance is used to calculate the label correlation.

3.4 Combining instance similarities

Equation 2 only considers the relationship between labels, and the potential relationship between instances is ignored. From [37, 38], Considering the dependency among instances, the distribution information of data samples can be retained to the maximum extent. Introducing the instance similarity regularization term $\boldsymbol{\Omega}(\boldsymbol{W}_1)$, Eq. 2 can be optimized as

$$\min_{\boldsymbol{W}_1,\boldsymbol{W}_2} \frac{1}{2}\left\|\tanh(\boldsymbol{X}\boldsymbol{W}_1)\boldsymbol{W}_2-\boldsymbol{Y}\right\|_F^2+\frac{\alpha}{2}tr(\boldsymbol{W}_2\boldsymbol{R}\boldsymbol{W}_2{}^T) + \beta\left\|\boldsymbol{W}_2\right\|_1+\frac{\lambda}{2}\boldsymbol{\Omega}(\boldsymbol{W}_1)+\frac{\gamma}{2}\left\|\boldsymbol{W}_1\right\|_F^2 \tag{3}$$

$\boldsymbol{\Omega}(\boldsymbol{W}_1)$ can be defined as

$$\Omega(\boldsymbol{W}_1)=\frac{1}{2}\sum_{i,j}\left\|\boldsymbol{W}_1{}^T x_i-\boldsymbol{W}_1{}^T\boldsymbol{x}_j\right\|_2^2\boldsymbol{S}_{ij}=tr((\boldsymbol{X}\boldsymbol{W}_1)^T\boldsymbol{L}\boldsymbol{X}\boldsymbol{W}_1) \tag{4}$$

where $\boldsymbol{S}_{ij}$ is the similarity between the i-th and j-th instances, $\boldsymbol{L}$ is the graph Laplacian matrix of the k-nearest neighbor graph $\boldsymbol{S}$, $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{S}$, $\boldsymbol{D}_{ii}=\sum_{j=1}^{n}\boldsymbol{S}_{ij}$, specifically, can be expressed as

$$\boldsymbol{S}_{ij}=\begin{cases} exp\left(-\frac{|x_i-x_j|^2}{\sigma^2}\right) & x_i\in\boldsymbol{N}_K(x_j)\ or\ x_j\in\boldsymbol{N}_K(x_i) \\ 0 & otherwise \end{cases} \tag{5}$$

From Eq. 5, if there is a strong similarity between x_i and x_j, then the distance between them will be smaller, otherwise, the distance between instances will be larger. Therefore, considering the instance similarity regularization term, i.e., minimization $\boldsymbol{\Omega}(\boldsymbol{W}_1)$ can be more accurately solved for the coefficient matrix $\boldsymbol{W}_1$, Eq. 4 can further be formulated as

$$f(\boldsymbol{W})=\min_{\boldsymbol{W}_1,\boldsymbol{W}_2}\frac{1}{2}\left\|\tanh(\boldsymbol{X}\boldsymbol{W}_1)\boldsymbol{W}_2-\boldsymbol{Y}\right\|_F^2+\frac{\alpha}{2}tr(\boldsymbol{W}_2\boldsymbol{R}\boldsymbol{W}_2{}^T)+\beta\left\|\boldsymbol{W}_2\right\|_1 +\frac{\lambda}{2}tr((\boldsymbol{X}\boldsymbol{W}_1)^T\boldsymbol{L}\boldsymbol{X}\boldsymbol{W}_1)+\frac{\gamma}{2}\left\|\boldsymbol{W}_1\right\|_F^2 \tag{6}$$

where α, β, λ, and γ are all positive constants, and their values are determined by five-fold cross-validation on the training data set.

3.5 Optimization of LLFN model

There are two model coefficients $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ to be optimized in Eq. 6. Obviously, it is very difficult to optimize them at the same time. Therefore, we use alternate optimization techniques to optimize $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$. Specifically, first, fix $\boldsymbol{W}_1$, use the accelerated proximal gradient method to optimize $\boldsymbol{W}_2$, then fix $\boldsymbol{W}_2$, use the gradient descent algorithm to optimize $\boldsymbol{W}_1$, and finally obtain the optimal $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$.

1. Fix $\boldsymbol{W}_1$, update $\boldsymbol{W}_2$

When $\boldsymbol{W}_1$ is fixed, the objective function of optimizing $\boldsymbol{W}_2$ can be further written as

$$\min_{\boldsymbol{W}_1,\boldsymbol{W}_2}\frac{1}{2}\left\|\tanh(\boldsymbol{X}\boldsymbol{W}_1)\boldsymbol{W}_2-\boldsymbol{Y}\right\|_F^2+\frac{\alpha}{2}tr(\boldsymbol{W}_2\boldsymbol{R}\boldsymbol{W}_2{}^T)+\beta\left\|\boldsymbol{W}_2\right\|_1 \tag{7}$$

It can be seen that Solving $\boldsymbol{W}_2$ in problem Eq. 7 is a convex optimization problem, but since the learning objective $\boldsymbol{W}_2$ of the model in this paper with l_1-norm regularization term, resulting in $\boldsymbol{W}_2$ is non-smooth and cannot be solved directly by deriving the derivative. Therefore, according to the literature [41], this paper uses Accelerated Proximal Gradient (APG) to solve the model parameters $\boldsymbol{W}_2$.

The convex optimization problem is generally divided into two parts by APG, and the equation is expressed as follows

$$\min_{\boldsymbol{W}_2\in H}F(\boldsymbol{W}_2)=f(\boldsymbol{W}_2)+g(\boldsymbol{W}_2) \tag{8}$$

where H denotes the Hilbert space, $f(\boldsymbol{W}_2)$ is a smooth convex function and $g(\boldsymbol{W}_2)$ is a non-smooth convex function. For $f(\boldsymbol{W}_2)$ satisfying the Lipschitz condition, then for any matrix $\boldsymbol{W}_{2_1}$ and $\boldsymbol{W}_{2_2}$ have

$$\left\|\nabla f(\boldsymbol{W}_{2_1}) - \nabla f(\boldsymbol{W}_{2_2})\right\| \le L_f \left\|\Delta \boldsymbol{W}_2\right\| \tag{9}$$

where L_f is the Lipschitz constant,$\Delta \boldsymbol{W}_2 = \boldsymbol{W}_{2_1} - \boldsymbol{W}_{2_2}$. In accelerated gradient descent it is necessary to introduce $Q\left(\boldsymbol{W}_2, \boldsymbol{W}_2^{(t)}\right)$ to quadratic approximation $F\left(\boldsymbol{W}_2\right)$, instead of direct minimization $F\left(\boldsymbol{W}_2\right)$, $Q\left(\boldsymbol{W}_2, \boldsymbol{W}_2^{(t)}\right)$defined as

$$\begin{aligned} Q(\boldsymbol{W}_2, \boldsymbol{W}_2{}^{(t)}) &= f(\boldsymbol{W}_2{}^{(t)}) \\ &+ \langle \nabla f(\boldsymbol{W}_2{}^{(t)}), \boldsymbol{W}_2 - \boldsymbol{W}_2{}^{(t)} \rangle + \frac{L_f}{2}\left\|\boldsymbol{W}_2 - \boldsymbol{W}_2{}^{(t)}\right\|_F^2 + g(\boldsymbol{W}_2) \end{aligned} \tag{10}$$

When

$$\boldsymbol{G}^{(t)} = \boldsymbol{W}_2{}^{(t)} - \frac{1}{L_f}\nabla f(\boldsymbol{W}_2{}^{(t)}) \tag{11}$$

Then Eq. 10 can be written as

$$\boldsymbol{W}_2 = \arg\min_{\boldsymbol{W}_2} Q(\boldsymbol{W}_2, \boldsymbol{W}_2{}^{(t)}) = \arg\min_{\boldsymbol{W}_2} \mathrm{g}(\boldsymbol{W}_2) + \frac{L_f}{2}\left\|\boldsymbol{W}_2 - \boldsymbol{G}^{(t)}\right\|_F^2 \tag{12}$$

From Eqs. 7 and 8, $f\left(\boldsymbol{W}_2\right)$ and $g\left(\boldsymbol{W}_2\right)$ are further expressed as

$$\begin{aligned} f(\boldsymbol{W}_2) = &\frac{1}{2}\left\|\tanh(\boldsymbol{X}\boldsymbol{W}_1)\boldsymbol{W}_2 - Y\right\|_F^2 + \frac{\alpha}{2}tr(\boldsymbol{W}_2\boldsymbol{R}\boldsymbol{W}_2{}^T) \\ &+ \frac{\lambda}{2}tr((\boldsymbol{X}\boldsymbol{W}_1)^T\boldsymbol{L}\boldsymbol{X}\boldsymbol{W}_1) + \frac{\gamma}{2}\left\|\boldsymbol{W}_1\right\|_F^2 \end{aligned} \tag{13}$$

$$g(\boldsymbol{W}_2) = \beta\left\|\boldsymbol{W}_2\right\|_1 \tag{14}$$

Then according to Eqs. 12, 13, and 14 coefficient matrix $\boldsymbol{W}_2$ can be optimized by

$$\boldsymbol{W}_2 = \arg\min_{\boldsymbol{W}_2} \frac{1}{2}\left\|\boldsymbol{W}_2 - \boldsymbol{G}^{(t)}\right\|_F^2 + \frac{\beta}{L_f}\left\|\boldsymbol{W}_2\right\|_1 \tag{15}$$

In [42], let $\boldsymbol{W}_2^{(t)} = \boldsymbol{W}_{2_t} + \frac{b_{t-1}-1}{b_t}(\boldsymbol{W}_{2_t} - \boldsymbol{W}_{2_{t-1}})$, $\boldsymbol{W}_{2_t}$ and $\boldsymbol{W}_{2_{t-1}}$ here are the coefficient matrices of the t-th and $t-1$-th iterations respectively. When the sequence b_t is satisfied $b_{t+1}^2 - b_{t+1} \le b_t^2$, the convergence rate of the algorithm can be increased to $O\left(t^{-2}\right)$. Since $g(\boldsymbol{W}_2)$ is l_1-norm, the iterative solution for $\boldsymbol{W}_2$ is as follows

$$\boldsymbol{W}_{2_{t+1}} = \boldsymbol{S}_\varepsilon\left[\boldsymbol{G}^{(t)}\right] = \arg\min_{\boldsymbol{W}_2} \varepsilon\left\|\boldsymbol{W}_2\right\|_1 + \frac{1}{2}\left\|\boldsymbol{W}_2 - \boldsymbol{G}^{(t)}\right\|_F^2 \tag{16}$$

where $\boldsymbol{S}_\varepsilon[\cdot]$ is the soft-threshold operator, for each element W_{ij} and $\varepsilon = \frac{\beta}{L_f}$ 0, the soft-threshold operator is defined as

$$S_\varepsilon\left[\boldsymbol{G}^{(t)}\right] = \begin{cases} w_{ij} - \varepsilon & \text{if } w_{ij} > \varepsilon \\ w_{ij} + \varepsilon & \text{if } w_{ij} < -\varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

Next, verify the Lipschitz continuity of Eq. 7, and according to Eq. 7, let $\boldsymbol{M} = \tanh\left(\boldsymbol{X}\boldsymbol{W}_1\right)$, $\nabla f\left(\boldsymbol{W}_2\right)$ is

$$\nabla f\left(\boldsymbol{W}_2\right) = \boldsymbol{M}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{W}_2 - \boldsymbol{M}^{\mathrm{T}}\boldsymbol{Y} + \alpha\boldsymbol{W}_2\boldsymbol{R} \tag{18}$$

Given $\boldsymbol{W}_{2_1}$ and $\boldsymbol{W}_{2_2}$, we obtain

$$\begin{aligned} \left\|\nabla f(\boldsymbol{W}_{2_1}) - \nabla f(\boldsymbol{W}_{2_2})\right\|_F^2 &= \left\|\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{W}_2 - \alpha\boldsymbol{W}_2\boldsymbol{R}\right\|_F^2 \\ &\le 2\left\|\boldsymbol{M}^T\boldsymbol{M}\Delta\boldsymbol{W}_2\right\|_F^2 + 2\left\|\alpha\Delta\boldsymbol{W}_2\boldsymbol{R}\right\|_F^2 \\ &\le 2\left\|\boldsymbol{M}^T\boldsymbol{M}\right\|_2^2\left\|\Delta\boldsymbol{W}_2\right\|_F^2 + 2\left\|\alpha\boldsymbol{R}\right\|_2^2\left\|\Delta\boldsymbol{W}_2\right\|_F^2 \\ &= \left(2\left\|\boldsymbol{M}^T\boldsymbol{M}\right\|_2^2 + 2\left\|\alpha\boldsymbol{R}\right\|_2^2\right)\left\|\Delta\boldsymbol{W}_2\right\|_F^2 \\ &= \left(2\delta_{\max}^2(\boldsymbol{M}^T\boldsymbol{M}) + 2\delta_{\max}^2(\alpha\boldsymbol{R})\right)\left\|\Delta\boldsymbol{W}_2\right\|_F^2 \end{aligned} \tag{19}$$

where $\Delta\boldsymbol{W}_2 = \boldsymbol{W}_{2_1} - \boldsymbol{W}_{2_2}$, $\delta_{\max}(\cdot)$ is the maximum value of singularity of the given matrix. In summary, we can get

$$\begin{aligned} \left\|\nabla f\left(\boldsymbol{W}_{2_1}\right) - \nabla f\left(\boldsymbol{W}_{2_2}\right)\right\|_F^2 \le &\left(2\delta_{\max}^2\left(\boldsymbol{M}^T\boldsymbol{M}\right)\right. \\ &\left.+2\delta_{\max}^2(\alpha\boldsymbol{R})\right)\left\|\Delta\boldsymbol{W}_2\right\|_F^2 \end{aligned} \tag{20}$$

In short, the Lipschitz constant is

$$L_f = \sqrt{2\delta_{\max}^2\left(\boldsymbol{M}^T\boldsymbol{M}\right) + 2\delta_{\max}^2(\alpha\boldsymbol{R})} \tag{21}$$

2. Fix $\boldsymbol{W}_2$, update $\boldsymbol{W}_1$

When $\boldsymbol{W}_2$ is fixed, the objective function of updating $\boldsymbol{W}_1$ is written as

$$\begin{aligned} \min_{\boldsymbol{W}_1, \boldsymbol{W}_2} &\frac{1}{2}\left\|\tanh(\boldsymbol{X}\boldsymbol{W}_1)\boldsymbol{W}_2 - \boldsymbol{Y}\right\|_F^2 + \frac{\lambda}{2}tr((\boldsymbol{X}\boldsymbol{W}_1)^T\boldsymbol{L}\boldsymbol{X}\boldsymbol{W}_1) \\ &+ \frac{\gamma}{2}\left\|\boldsymbol{W}_1\right\|_F^2 \end{aligned} \tag{22}$$

The gradient descent algorithm is used to solve for $\boldsymbol{W}_1$, and the derivative of $\boldsymbol{W}_1$ for the above equation can be obtained as

$$\begin{aligned} \nabla f\left(\boldsymbol{W}_1\right) = &\boldsymbol{X}^T(\boldsymbol{1} - \boldsymbol{M}\odot\boldsymbol{M})\left(\boldsymbol{M}\boldsymbol{W}_2\boldsymbol{W}_2^T - 2\boldsymbol{Y}\boldsymbol{W}_2^T\right) \\ &+ \lambda\boldsymbol{X}^T\boldsymbol{L}\boldsymbol{X}\boldsymbol{W}_1 + \gamma\boldsymbol{W}_1 \end{aligned} \tag{23}$$

Where $\odot$ is the Hadamard product operator, then the updated $\boldsymbol{W}_1$ is

$$\boldsymbol{W}_1 = \boldsymbol{W}_1 - \eta\nabla f(\boldsymbol{W}_1) \tag{24}$$

Based on the above iterative optimization process, the specific iterative solution procedure is summarized in Algorithm 1.

Algorithm 1: Neural network-specific feature learning algorithm based on accelerated approximate gradient descent method

Input: Training data matrix: $\boldsymbol{X} \in \mathbb{R}^{n\times p}$, label matrix: $\boldsymbol{Y} \in \mathbb{R}^{n\times l}$, weight parameters:$\alpha, \beta, \gamma, \lambda, \mu$;
Output: Model coefficient matrix: $\boldsymbol{W}_1 \in \mathbb{R}^{p\times d}, \boldsymbol{W}_2 \in \mathbb{R}^{d\times l}$;
Initialization:
$b_0, b_1 \leftarrow 1, t \leftarrow 1, \boldsymbol{W}_{2_0}, \boldsymbol{W}_{2_1} \leftarrow (\boldsymbol{X}^T\boldsymbol{X} + \mu\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{Y}, \boldsymbol{W}_1 \leftarrow rand(p, d)$;
Calculation of the label correlation matrix $\boldsymbol{R}$ using Hamming distances;
Computation of the Laplace matrix $\boldsymbol{L}$ using the K-nearest neighbor probability map model;
Calculation L_f according to Eq.21;
while not converged **do**
  $\boldsymbol{W_2}^{(t)} \leftarrow \boldsymbol{W}_{2_t} + \frac{b_{t-1}-1}{b_t}\left(\boldsymbol{W}_{2_t} - \boldsymbol{W}_{2_{t-1}}\right)$;
  $\boldsymbol{G}^{(t)} \leftarrow \boldsymbol{W}_2{}^{(t)} - \frac{1}{L_f}\nabla f(\boldsymbol{W}_2{}^{(t)})$;
  $\boldsymbol{W}_{t+1} \leftarrow S_{\frac{\beta}{L_f}}\left(\boldsymbol{G}^{(t)}\right)$;
  $b_{t+1} \leftarrow \frac{1+\sqrt{4b_t^2+1}}{2}$;
  $\boldsymbol{W}_2 \leftarrow \boldsymbol{W}_{2_t}$;
  Update $\boldsymbol{W}_1$ by Eq.24;
  $t \leftarrow t + 1$;

Algorithm 2: Test of LLFN

Input: Testing data matrix: $\boldsymbol{X}_{te} \in \mathbb{R}^{m\times p}$, model coefficient matrix: $\boldsymbol{W}_2 \in \mathbb{R}^{d\times l}$, threshold: τ;
Output: Predictive label matrix: $\boldsymbol{Y}_{te}$, score matrix: $\boldsymbol{S}_{te}$;
$\boldsymbol{S}_{te} \leftarrow \tanh(\boldsymbol{X}_{te}\boldsymbol{W}_1)\boldsymbol{W}_2$;
$\boldsymbol{Y}_{te} \leftarrow \text{sign}\,(\boldsymbol{S}_{te} - \tau)$.

Algorithm 3: LLFN-SVM Method

Input: Testing data matrix: $\boldsymbol{X}_{te} \in \mathbb{R}^{m\times p}$, Binary classifier: SVM, Kernel function: linear;
Output: Predictive label matrix: $\boldsymbol{Y}_{te}$, score matrix: $\boldsymbol{S}_{te}$;
Training:
Learning the model coefficient matrix $\boldsymbol{W}_1, \boldsymbol{W}_2$ of LLFN by Algorithm 1;
for $i = 1$ to l **do**
  $\text{id}_i \leftarrow find(\boldsymbol{W}_{2_i} \neq 0)$;
  $\boldsymbol{X}_{tr}^i \leftarrow \boldsymbol{X}(:, \text{id}_i)$;
  $h_i \leftarrow f(\boldsymbol{X}_{tr}^i, \boldsymbol{Y}, \theta)$;
testing:
for $i = 1$ to l **do**
  $\boldsymbol{X}_{te}^i \leftarrow \boldsymbol{X}_{te}^i(:, \text{id}_i)$;
  $\{\boldsymbol{Y}_{te}^i, \boldsymbol{S}_{te}^i\} \leftarrow h_i(\boldsymbol{X}_{te}^i, \boldsymbol{Y}_{te}, \theta)$;
$\boldsymbol{Y}_{te} \leftarrow \left[\boldsymbol{Y}_{\text{te}}^1, \boldsymbol{Y}_{\text{te}}^2 \ldots, \boldsymbol{Y}_{\text{te}}^l\right]$;
$\boldsymbol{S}_{test} \leftarrow \left[\boldsymbol{S}_{\text{te}}^1, \boldsymbol{S}_{\text{te}}^2 \ldots, \boldsymbol{S}_{\text{te}}^l\right]$;

The nonzero entities W_{2_i} are considered as label-specific features of y_i, which are used as inputs to the classification algorithm with multi-labels, and then the binary classifier BSVM is used to achieve multi-label classification. The procedure is summarized in Algorithm 3.

3.6 Complexity analysis

The time complexity of LLFN consists of two main components: the algorithm initialization and the iterative process. The complexity of updating the weight matrix $\boldsymbol{W}_{2_0}$ of the model in initialization is $O(np^2 + npl + p^3 + p^2l)$, the complexity of the computing label similarity matrix is

Table 1 Description of the LLFN datasets

Data set	Instances	Features	Labels	Cardinality	Domains
Arts[2]	5000	462	26	1.636	Text
Computers[2]	5000	681	33	1.508	Text
Education[2]	5000	550	33	1.461	Text
Emotion[1]	593	72	6	1.869	Music
Image[3]	2000	294	5	1.236	Image
Medical [1]	978	1449	45	1.245	Text
Science[2]	5000	743	40	1.451	Text
Social[2]	5000	1047	39	1.233	Text
Health[2]	5000	612	32	1.663	Text
Society[2]	5000	636	27	1.461	Text
Business[2]	5000	438	30	1.588	Text
Recreation[2]	5000	606	22	1.423	Text

$O(nl^2)$, and the graph Laplacian matrix $\boldsymbol{L}$ requires $O(n^2d)$. During the iteration, the complexity of computing the Lipschitz constant L_f is $O(npd + nd^2 + d^3 + nl^2 + l^3)$, and the focus in the loop process is to compute $\nabla f(\boldsymbol{W_1})$. Which is obtained from Eq. 23 as $O(npd + nd^2 + d^2l + ndl + dl^2)$. In summary, the complexity of $\nabla f(\boldsymbol{W}_1)$ and L_f is relatively the highest time complexity, and if the time complexity of the initialization process is relatively low, then the overall complexity has to be the higher order of magnitude part of the time complexity. Furthermore, because L_f only needs to be calculated once, so the complexity of the whole algorithm is $O(npd + nd^2 + d^2l + ndl + dl^2)$. Meanwhile, we also compare the time complexity of the LLFN algorithm with LLSF, LSML, JLCLS, and BDLS algorithms. From the work in [16, 34, 43, 44], it can be seen that the time complexity of LLSF is $O\left(d^2 + dl + l^2 + nd + nl\right)$, the complexity of LSML is $O\left((n + l)d^2 + (n + d)l^2 + dnl + l^3 + d^3\right)$, the complexity of JLCLS is $O\left((n + 1)\left(d^2l^2 + nl^2 + nd^2l\right) + d^3 + l^3\right)$, and the complexity of BDLS is $O((n + d + l)ldt)$. The comparison finds that the algorithm proposed in this paper is competitive with other algorithms in terms of time efficiency.

4 Experiment

In this section, to verify the competitiveness and extensiveness of our proposed LLFN, six existing multi-label classification algorithms are used to compare with LLFN, and these methods are experimented on 12 datasets using five multi-label evaluation criteria. The dataset analysis, performance metrics, and comparison algorithms are first briefly introduced to prepare for the analysis of the experimental results.

4.1 Data sets

In this section, the comparison data were selected from 12 multi-label datasets of different domains, and the details of the experimental datasets are described in Table 1. Specifically, These datasets can be downloaded from Mulan,[1] Yahoo,[2] and Image.[3]

4.2 Evaluation metrics

In contrast to single-label learning, multi-label learning is not unique due to the number of labels corresponding to the samples to be classified. The classification complexity leads to the complexity of measuring the performance of multi-label generalization, while the goodness of label prediction can be measured based on certain evaluation metrics. To measure the performance of multi-label classification and feature selection intuitively and numerically, five evaluation metrics [6] commonly used in the multi-label domain are selected in this paper to compare with the algorithms introduced above. Among them, the $D = \{(X_{it}, Y_{il} | 1 \le t \le p, 1 \le i \le n, 1 \le l \le L)\}$ is the multi-label data set.

- *Hamming Loss (HL* $\downarrow$*)* evaluates the variance between the set of true label sets and the predicted label set that is the number of times a sample label pair is misclassified.

$$Hamming\ Loss = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{|Y|}\left|h(x_i)\Delta Y_i\right|\right) \tag{25}$$

- *Average Precision (AP* $\uparrow$*)* is used to evaluate the average score of the real labels ranked higher than the non-real labels in the predicted label ranking of the whole sample.

$$Average\ Precision = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|}\cdot \sum_{y\in Y_i}\frac{\left|\left\{y' \mid \mathrm{rank}_f(x_i, y') \le \mathrm{rank}_f(x_i, y), y' \in Y_i\right\}\right|}{\mathrm{rank}_f(x_i, y)} \tag{26}$$

- *Ranking Loss (RL* $\downarrow$*)* indicates the probability value that the confidence level of the associated labels in the sample prediction result is smaller than the confidence level of the unassociated labels.

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

[2] code: http://www.kecl.ntt.co.jp/as/members/ueda/yahoo.tar.

[3] code: http://cse.seu.edu.cn/people/zhangml/Resources.htm#data.

$$RankingLoss = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i||\overline{Y}_i|}\cdot\left|\{(y_1,y_2)|f(x_i,y_1) \le f(x_i,y_2),(y_1,y_2)\in Y_i\times\overline{Y}_l\}\right| \quad (27)$$

- *One-Error (OE ↓)* reflects the probability that the Top-Ranked Label in the prediction result is not in the true set of labels for that sample.

$$One\ Error = \frac{1}{n}\sum_{1}^{n}\left[\left[\underset{y\in Y}{\operatorname{argmax}} f(x_i,y)\notin Y_i\right]\right] \quad (28)$$

- *Coverage (CV ↓)* is used to evaluate the ranking of the marks to be tested for all samples, and how many steps are needed to cover all the marks related to the sample on average.

$$Coverage = \frac{1}{n}\sum_{i=1}^{n}\max_{y\in Y_i}\operatorname{rank}_f(x_i,y)-1 \quad (29)$$

4.3 Comparative algorithms

- *ML-kNN* [8] It is based on the classical KNN method for multi-label data, which counts the number of occurrences of these neighboring instances to be predicted, and the maximum a posteriori probability (MAP) principle is used to identify the label set of the unknown sample. In our experiments, the parameter k is set to 10.
- *LIFT* [15] It uses clustering techniques to study the positive and negative instances of each category label to construct label-specific features. Then the generated label-specific features are then used to generalize a binary classification model for the corresponding category labels. LIFT reduces the dimensionality of the feature space but does not consider label correlation. The ratio parameter r is set to 0.1 for all data sets.
- *LLSF* [16] A method of sparse superposition is used to learn related feature subsets for label-specific feature extraction, but does not consider instance correlation. The parameters α, β, and γ are set to 0.1, 0.1 and 0.01 respectively. the threshold τ is set to 0.5.
- *LSML* [34] It handles missing multi-label specific data for classification by learning higher-order label correlation matrix with label feature method. The parameters λ_1,λ_2, λ_3, and λ_4 are set to 10^2, 10^{-5}, 10^{-3}, and 10^{-5} respectively.

Table 2 Experimental results (mean ± std) of the comparison algorithm on HL (↓)

Data set	LLFN	LLFN-BSVM	LLSF	LSML	KNN	LIFT	JLCLS	BDLS
Arts	0.0543 ± 0.0018	0.0531 ± 0.0010	0.0580 ± 0.0009	0.0664 ± 0.0017	0.0599 ± 0.0009	**0.0530 ± 0.0015**	0.0558 ± 0.0013	0.0559 ± 0.0010
Computers	0.0338 ± 0.0010	0.0333 ± 0.0012	0.0378 ± 0.0018	0.0386 ± 0.0007	0.0386 ± 0.0014	**0.0330 ± 0.0014**	0.0359 ± 0.0009	0.0381 ± 0.0012
Education	0.0377 ± 0.0006	**0.0371 ± 0.0012**	0.0425 ± 0.0012	0.0477 ± 0.0010	0.0383 ± 0.0004	0.0372 ± 0.0006	0.0386 ± 0.0003	0.0399 ± 0.0006
Emotion	0.1926 ± 0.0104	0.1880 ± 0.0150	0.1982 ± 0.0039	0.2015 ± 0.0089	0.2020 ± 0.0120	**0.1866 ± 0.0087**	0.2583 ± 0.0068	0.1880 ± 0.0086
Image	0.1807 ± 0.0074	0.1897 ± 0.0062	0.2150 ± 0.0046	0.2047 ± 0.0158	0.1745 ± 0.0074	**0.1564 ± 0.0028**	0.2045 ± 0.0051	0.1917 ± 0.0052
Medical	0.0105 ± 0.0016	**0.0102 ± 0.0011**	0.0103 ± 0.0010	0.0110 ± 0.0012	0.0156 ± 0.0006	0.0122 ± 0.0010	0.0192 ± 0.0010	0.0106 ± 0.0005
Science	0.0314 ± 0.0011	**0.0307 ± 0.0012**	0.0350 ± 0.0012	0.0390 ± 0.0018	0.0326 ± 0.0011	0.0309 ± 0.0006	0.0330 ± 0.0003	0.0354 ± 0.0012
Social	0.0205 ± 0.0005	**0.0191 ± 0.0009**	0.0356 ± 0.0078	0.0240 ± 0.0012	0.0210 ± 0.0007	0.0196 ± 0.0006	0.0218 ± 0.0005	0.0256 ± 0.0007
Health	0.0340 ± 0.0006	**0.0310 ± 0.0010**	0.0415 ± 0.0025	0.0352 ± 0.0015	0.0450 ± 0.0007	0.0312 ± 0.0008	0.0362 ± 0.0006	0.0328 ± 0.0007
Society	0.0522 ± 0.0010	**0.0510 ± 0.0022**	0.0567 ± 0.0014	0.0647 ± 0.0027	0.0545 ± 0.0018	0.0511 ± 0.0013	0.0533 ± 0.0009	0.0548 ± 0.0008
Business	0.0259 ± 0.0012	**0.0244 ± 0.0005**	0.0513 ± 0.0076	0.0250 ± 0.0009	0.0265 ± 0.0015	0.0244 ± 0.0007	0.0280 ± 0.0007	0.0277 ± 0.0007
Recreation	0.0535 ± 0.0007	**0.0525 ± 0.0010**	0.0551 ± 0.0006	0.0677 ± 0.0021	0.0602 ± 0.0012	0.0529 ± 0.0022	0.0573 ± 0.0008	0.0554 ± 0.0014

Table 3 Experimental results (mean ± std) of the comparison algorithm on AP (↑)

Data set	LLFN	LLFN-BSVM	LLSF	LSML	KNN	LIFT	JLCLS	BDLS
Arts	0.6287 ± 0.0050	**0.6360 ± 0.0119**	0.6085 ± 0.0081	0.6286 ± 0.0102	0.5267 ± 0.0093	0.6246 ± 0.0076	0.6299 ± 0.0083	0.6233 ± 0.0098
Computers	0.7142 ± 0.3063	**0.7190 ± 0.0078**	0.6926 ± 0.0110	0.7132 ± 0.0084	0.6446 ± 0.0156	0.7083 ± 0.0029	0.7048 ± 0.0055	0.6861 ± 0.0097
Education	0.6394 ± 0.0031	**0.6493 ± 0.0121**	0.6079 ± 0.0070	0.6374 ± 0.0131	0.6101 ± 0.0098	0.6383 ± 0.0140	0.6385 ± 0.0064	0.6339 ± 0.0124
Emotion	0.8103 ± 0.0163	**0.8210 ± 0.0191**	0.8079 ± 0.0093	0.8022 ± 0.0118	0.7965 ± 0.0116	0.8208 ± 0.0072	0.7624 ± 0.0075	0.8129 ± 0.0137
Image	0.7875 ± 0.0185	0.7727 ± 0.0170	0.7758 ± 0.0122	0.7789 ± 0.0118	0.7895 ± 0.0124	**0.8214 ± 0.0107**	0.7615 ± 0.0154	0.7782 ± 0.0132
Medical	**0.9075 ± 0.0182**	0.8987 ± 0.0145	0.8980 ± 0.0190	0.8996 ± 0.0185	0.8041 ± 0.0251	0.8709 ± 0.0163	0.8498 ± 0.0148	0.9026 ± 0.0114
Science	0.6134 ± 0.0129	**0.6181 ± 0.0129**	0.5790 ± 0.0097	0.6087 ± 0.0122	0.5556 ± 0.0077	0.6069 ± 0.0088	0.6081 ± 0.0053	0.6078 ± 0.0122
Social	0.7819 ± 0.0062	0.7844 ± 0.0129	0.7612 ± 0.0108	0.7786 ± 0.0104	0.7594 ± 0.0107	**0.7881 ± 0.0081**	0.7778 ± 0.0086	0.7568 ± 0.0071
Health	0.7848 ± 0.0044	**0.8034 ± 0.0076**	0.7636 ± 0.0071	0.7868 ± 0.0057	0.6931 ± 0.0007	0.8022 ± 0.0096	0.7854 ± 0.0061	0.8012 ± 0.0006
Society	0.6371 ± 0.0117	**0.6523 ± 0.0043**	0.6283 ± 0.0117	0.6422 ± 0.0064	0.6071 ± 0.0018	0.6439 ± 0.0097	0.6381 ± 0.0178	0.6269 ± 0.0076
Business	0.8832 ± 0.0057	**0.8932 ± 0.0042**	0.8645 ± 0.0061	0.8876 ± 0.0052	0.8831 ± 0.0122	0.8925 ± 0.0049	0.8861 ± 0.0045	0.8774 ± 0.0066
Recreation	0.6426 ± 0.0169	**0.6501 ± 0.0084**	0.6347 ± 0.0081	0.6434 ± 0.0104	0.4807 ± 0.0038	0.6448 ± 0.0131	0.6463 ± 0.0056	0.6383 ± 0.0108

Table 4 Experimental results (mean ± std) of the comparison algorithm on OE (↓)

Data set	LLFN	LLFN-BSVM	LLSF	LSML	KNN	LIFT	JLCLS	BDLS
Arts	0.4510 ± 0.0050	**0.4458 ± 0.0179**	0.4636 ± 0.0081	0.4466 ± 0.0149	0.6090 ± 0.0133	0.4590 ± 0.0104	0.4576 ± 0.0173	0.4696 ± 0.0198
Computers	**0.3402 ± 0.0118**	0.3416 ± 0.0148	0.3570 ± 0.0156	0.3458 ± 0.0140	0.4292 ± 0.0175	0.3486 ± 0.0062	0.3628 ± 0.0065	0.3992 ± 0.0109
Education	0.4584 ± 0.0100	**0.4550 ± 0.0207**	0.4800 ± 0.0084	0.4590 ± 0.0198	0.5050 ± 0.0179	0.4662 ± 0.0210	0.4706 ± 0.0094	0.4800 ± 0.0191
Emotion	0.2411 ± 0.0376	0.2401 ± 0.0341	0.2479 ± 0.0107	0.2697 ± 0.0362	0.2900 ± 0.0209	**0.2243 ± 0.0154**	0.3288 ± 0.0164	0.2495 ± 0.0164
Image	0.3320 ± 0.0274	0.3485 ± 0.0235	0.3465 ± 0.0174	0.3415 ± 0.0229	0.3255 ± 0.0180	**0.2730 ± 0.0185**	0.3501 ± 0.0148	0.3410 ± 0.2610
Medical	0.1309 ± 0.0189	**0.1299 ± 0.0202**	0.1400 ± 0.0277	0.1391 ± 0.0319	0.2516 ± 0.0218	0.1647 ± 0.0185	0.2025 ± 0.0259	0.1370 ± 0.0267
Science	0.4774 ± 0.0117	**0.4694 ± 0.0136**	0.5010 ± 0.0135	0.4774 ± 0.0122	0.5514 ± 0.0090	0.4850 ± 0.0113	0.4918 ± 0.0093	0.5012 ± 0.0114
Social	0.2740 ± 0.0105	**0.2624 ± 0.0180**	0.2724 ± 0.0139	0.2732 ± 0.0081	0.3122 ± 0.0136	0.2660 ± 0.0108	0.2846 ± 0.0139	0.3202 ± 0.0126
Health	0.2518 ± 0.0063	**0.2336 ± 0.0093**	0.2560 ± 0.0146	0.2504 ± 0.0105	0.3890 ± 0.0180	0.2352 ± 0.0099	0.2600 ± 0.0064	0.2533 ± 0.0103
Society	0.3966 ± 0.0140	**0.3754 ± 0.0136**	0.3952 ± 0.0147	0.3844 ± 0.0111	0.4412 ± 0.0204	0.3870 ± 0.0149	0.4070 ± 0.0255	0.4246 ± 0.0100
Business	0.1148 ± 0.0091	**0.1052 ± 0.0056**	0.1232 ± 0.0050	0.1206 ± 0.0057	0.1164 ± 0.0125	0.1056 ± 0.0046	0.1202 ± 0.0059	0.1266 ± 0.0105
Recreation	0.4486 ± 0.0179	**0.4374 ± 0.0108**	0.4458 ± 0.0105	0.4586 ± 0.0119	0.6676 ± 0.0073	0.4460 ± 0.0177	0.4508 ± 0.0082	0.4640 ± 0.0116

Table 5 Experimental results (mean ± std) of the comparison algorithm on RL (↓)

Data set	LLFN	LLFN-BSVM	LLSF	LSML	KNN	LIFT	JLCLS	BDLS
Arts	0.1329 ± 0.0038	**0.1125 ± 0.0050**	0.1646 ± 0.0081	0.1325 ± 0.0060	0.1493 ± 0.0054	0.1148 ± 0.0039	0.1205 ± 0.0034	0.1148 ± 0.0020
Computers	0.0894 ± 0.0051	**0.0686 ± 0.0023**	0.1153 ± 0.0046	0.0889 ± 0.0017	0.0866 ± 0.0051	0.0695 ± 0.0029	0.0792 ± 0.0044	0.0771 ± 0.0041
Education	0.0999 ± 0.0028	**0.0700 ± 0.0028**	0.1447 ± 0.0079	0.1007 ± 0.0038	0.0783 ± 0.0031	0.0740 ± 0.0041	0.0840 ± 0.0037	0.1121 ± 0.0020
Emotion	0.1545 ± 0.0117	**0.1437 ± 0.0161**	0.1572 ± 0.0091	0.1591 ± 0.0093	0.1617 ± 0.0144	0.1471 ± 0.0109	0.1947 ± 0.0091	0.1518 ± 0.0141
Image	0.1742 ± 0.0156	0.1745 ± 0.0172	0.1837 ± 0.0104	0.1802 ± 0.0093	0.1750 ± 0.0148	**0.1470 ± 0.0097**	0.1813 ± 0.0154	0.1825 ± 0.0114
Medical	0.0161 ± 0.0058	0.0213 ± 0.0027	0.0227 ± 0.0066	**0.0159 ± 0.0041**	0.0442 ± 0.0118	0.0266 ± 0.0020	0.0221 ± 0.0070	0.0155 ± 0.0031
Science	0.1179 ± 0.0081	**0.0954 ± 0.0034**	0.1604 ± 0.0040	0.1214 ± 0.0057	0.1117 ± 0.0044	0.0972 ± 0.0021	0.1042 ± 0.0050	0.1023 ± 0.0051
Social	0.0592 ± 0.0028	0.0522 ± 0.0066	0.0871 ± 0.0035	0.0660 ± 0.0073	0.0524 ± 0.0042	**0.0479 ± 0.0022**	0.0553 ± 0.0019	0.0504 ± 0.0036
Health	0.0614 ± 0.0032	0.0402 ± 0.0030	0.0806 ± 0.0037	0.0601 ± 0.0035	0.0595 ± 0.0007	**0.0388 ± 0.0034**	0.0532 ± 0.0028	0.0551 ± 0.0045
Society	0.1384 ± 0.0071	0.1214 ± 0.0034	0.1614 ± 0.0110	0.1449 ± 0.0032	0.1338 ± 0.0052	**0.1203 ± 0.0066**	0.1260 ± 0.0074	0.1228 ± 0.0051
Business	0.0428 ± 0.0029	**0.0318 ± 0.0021**	0.0496 ± 0.0037	0.0406 ± 0.0023	0.0376 ± 0.0045	0.0341 ± 0.0032	0.0369 ± 0.0023	0.0404 ± 0.0032
Recreation	0.1439 ± 0.0068	0.1245 ± 0.0032	0.1610 ± 0.0066	0.1428 ± 0.0062	0.1835 ± 0.0022	**0.1211 ± 0.0057**	0.1277 ± 0.0051	0.1237 ± 0.0052

Table 6 Experimental results (mean ± std) of the comparison algorithm on CV (↓)

Data set	LLFN	LLFN-BSVM	LLSF	LSML	KNN	LIFT	JLCLS	BDLS
Arts	0.2042 ± 0.0080	**0.1718 ± 0.0070**	0.2379 ± 0.0081	0.2044 ± 0.0121	0.2069 ± 0.0049	0.1748 ± 0.0058	0.1869 ± 0.0059	0.1790 ± 0.0048
Computers	0.1320 ± 0.0088	**0.1048 ± 0.0040**	0.1582 ± 0.0054	0.1312 ± 0.0041	0.1256 ± 0.0074	0.1062 ± 0.0030	0.1182!'À ± 0.0070	0.1173 ± 0.0054
Education	0.1475 ± 0.0049	**0.0991 ± 0.0034**	0.1992 ± 0.0118	0.1485 ± 0.0057	0.1055 ± 0.0021	0.1034 ± 0.0052	0.1229 ± 0.0058	0.1151 ± 0.0041
Emotion	0.2949 ± 0.0141	**0.2822 ± 0.0085**	0.2987 ± 0.0104	0.2965 ± 0.0105	0.2954 ± 0.0178	0.2875 ± 0.0136	0.3311 ± 0.0108	0.2923 ± 0.0096
Image	0.1922 ± 0.0139	0.2059 ± 0.0142	0.1999 ± 0.0081	0.1993 ± 0.0118	0.193 ± 0.0148	**0.1716 ± 0.0097**	0.2003 ± 0.0095	0.1932 ± 0.0152
Medical	0.0272 ± 0.0090	0.0322 ± 0.0069	0.0333 ± 0.0058	**0.0255 ± 0.0049**	0.0627 ± 0.0148	0.0411 ± 0.0061	0.0336 ± 0.0098	0.0250 ± 0.0045
Science	0.1636 ± 0.0087	**0.1314 ± 0.0047**	0.2088 ± 0.0069	0.1675 ± 0.0103	0.1464 ± 0.0057	0.1344 ± 0.0026	0.1460 ± 0.0041	0.1523 ± 0.0044
Social	0.0875 ± 0.0035	0.0752 ± 0.0062	0.1212 ± 0.0072	0.0967 ± 0.0101	0.0729 ± 0.0050	**0.0696 ± 0.0024**	0.0814 ± 0.0018	0.0726 ± 0.0032
Health	0.1187 ± 0.0047	0.0803 ± 0.0055	0.1411 ± 0.0040	0.1174 ± 0.0064	0.1011 ± 0.0025	**0.0785 ± 0.0050**	0.1052 ± 0.0046	0.1051 ± 0.0054
Society	0.2189 ± 0.0099	**0.1881 ± 0.0052**	0.2471 ± 0.0130	0.2307 ± 0.0039	0.1999 ± 0.0058	0.1892 ± 0.0083	0.2018 ± 0.0255	0.1941 ± 0.0051
Business	0.0860 ± 0.0033	**0.0676 ± 0.0011**	0.0903 ± 0.0056	0.0819 ± 0.0035	0.0724 ± 0.0072	0.0706 ± 0.0028	0.0756 ± 0.0046	0.0783 ± 0.0022
Recreation	0.1939 ± 0.0078	0.1686 ± 0.0058	0.2111 ± 0.0078	0.1926 ± 0.0092	0.2231 ± 0.0030	**0.1641 ± 0.0091**	0.1742 ± 0.0076	0.1679 ± 0.0069

- *JLCLS* [43] It learns jointly by considering the mislabeled tags and tag-specific features. The algorithm uses alternating iterative optimization to obtain the completion matrix and label-specific features with full consideration of label correlation. The parameters α, β, and θ are searched in $\{2^{-10}, 2^{-9}, \ldots, 2^{9}, 2^{10}\}$, γ selects from $\{0.1, 1, 10\}$.
- *BDLS* [44] It considers bidirectional mapping and label causality and thereby learns specific features of the labels. The parameters α, β and λ are searched in $\{2^{-7}, 2^{-6}, \ldots, 2^{6}, 2^{7}\}$, γ selects from $\{0.01, 0, 1, 10\}$.
- *LLFN* The method proposed in this paper combines multi-label classification by neural networks after learning multi-label specific features, considering label relevance and instance relevance. The parameters $\alpha, \beta, \gamma, \lambda$, and η are searched in $\{2^{-10}, 2^{-9}, \ldots, 2^{9}, 2^{10}\}$, τ is also set to 0.5.
- *LLFN-BSVM* The binary classifier BSVM is added to LLFN, and a data matrix consisting of label-specific features generated by LLFN is set as the training data for BSVM. Where the kernel function is linear and all parameters are set the same as LLFN.

4.4 Experimental results

In order to accurately evaluate the performance of each multi-label classification algorithm, a five-fold cross-validation is applied to the training data of each dataset. The comparison of the values of the five evaluation metrics for each algorithm is shown in Tables 2, 3, 4, 5, 6, and the best results in the table are indicated by bold numbers. The evaluation metrics are followed by the symbols $\varepsilon \uparrow \varepsilon$ and $\varepsilon \downarrow \varepsilon$ after the evaluation metrics indicate that the larger the value of the evaluation metric is, the better the performance of the algorithm and the smaller the value is, the better the performance of the algorithm, respectively.

In addition, the Friedman test is used in this paper to compare the relative performance among the algorithms, and the corresponding critical values of the Friedman statistic and each evaluation metric are given in Table 7. At the significance level of $\alpha = 0.05$, the hypothesis that all algorithms have the same performance is explicitly rejected. Therefore, we need to use the Nemenyi test to further distinguish the classification performance of LLFN as well as other comparative algorithms on the 12 datasets. Figure 2 presents the CD plots for each algorithm under different evaluation metrics, respectively. In each subplot, if the corresponding mean ordinal values differ by at least the critical value domain (CD): $CD = q_\alpha \sqrt{\frac{K(K+1)}{6N}}$, then it indicates a significant difference in performance between classifiers. For the Nemenyi test, it can be calculated as $CD = 3.031 (K = 8, N = 12)$ at the significance level $\alpha = 0.05$ and the critical difference

Table 7 Summary of the Friedman statistics $F_F(K = 8, N = 12)$ and the critical value in each evaluation metric (K: Comparing Algorithms; N: Data Sets)

Metric	F_F	Critical value ($\alpha = 0.05$)
Hamming loss	20.4108	
Average precision	12.3432	
One error	15.9344	2.1310
Ranking loss	17.4975	
Coverage	13.8054	

$q_\alpha = 3.031$. As shown in Fig. 2, the algorithm with the red line connected to each subgraph is considered as the algorithm with less significant difference. To summarize the above experimental results, it can be concluded that:

1. Analyzing the optimal comparison experiments shown in Tables 2, 3, 4, 5, 6, it can be observed that LLFN-BSVM significantly outperforms the LLSF, LSML, KNN, JLCSC, and BDLS algorithms on the eight datasets in terms of HL metrics, while showing suboptimal results on art, computers, and emotion. And on AP and OE metrics, LLFN-BSVM presented optimal results on 10 data sets. For RL and CV metrics, LLFN-BSVM had the best experimental results with 6 and 7 datasets, respectively, and LLFN-BSVM slightly outperformed LIFT in RL and CV indicators in general. In addition, it was found from Fig. 2 that when the significance level $\alpha = 0.05$, LLFN-BSVM ranked first in all performance metrics. LLFN ranked higher than LLSF, LSML, KNN, JLCLS, and BDLS algorithms in HL, AP, and OE, but ranked just below JLCLS and BDLS algorithms in RL metrics and below BDLCS in CV algorithms. This verifies the effectiveness of the algorithm proposed in this paper, that is, the introduction of neural networks for label-specific feature learning can improve the performance of multi-label classification.
2. LLFN-BSVM performs better than LLFN in 85% of the cases and obtains more stable experimental results in comparison. Additionally, as shown in Tables 5 and 6, LLFN-BSVM and LIFT are close in RL and CV values and perform well. This is because the base classifiers of LLFN-BSVM and LIFT are SVM, and SVM classifiers cannot deal with multi-label problems directly but treat multi-label classification problems as multiple single-label classification problems, so they have superior results in RL and CV. In most cases, LLFN-BSVM has a better presentation on each performance metric compared to LIFT, which is because LIFT does not consider correlation information among labels and similar-

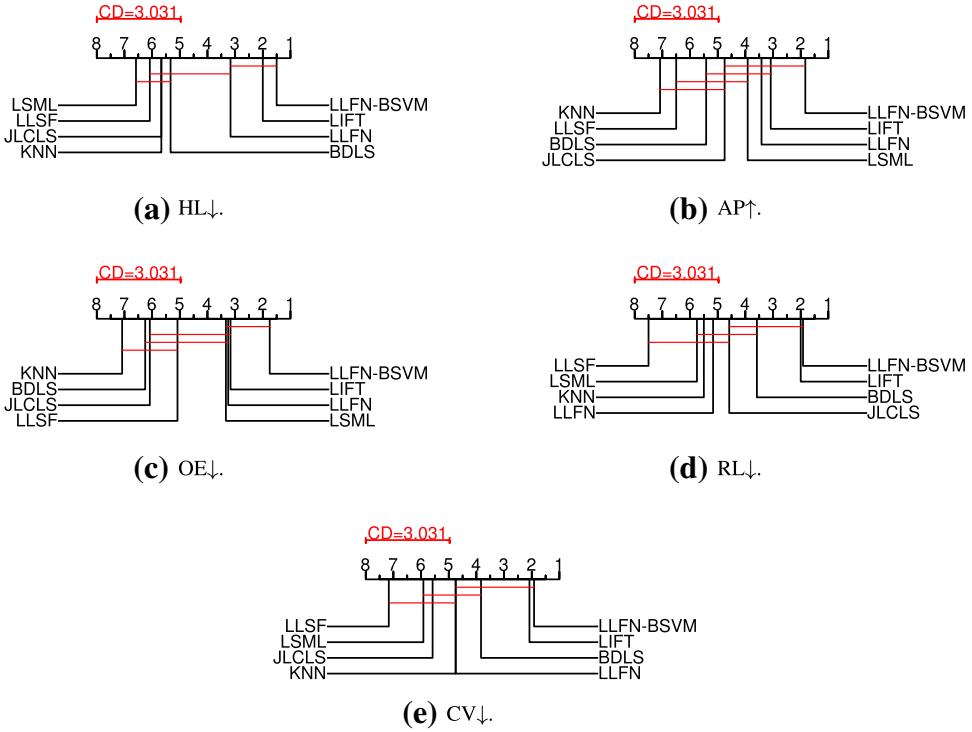


Fig. 2 Nemenyi test results for different evaluation metrics. (at $\alpha = 0.05$)

ity among instances, resulting in poor performance on the rest of the performance metrics.

3. We further observed that in most cases, for data sets with a larger number of samples (such as education, science, business, etc.), the neural network has higher accuracy for multi-label classification. For the image, medical, and social data sets (the cardinality of the average value of labels is about 1.2), the labels are relatively sparse compared to other data sets, resulting in insufficient label correlation information obtained from the original label set, so the least square loss model is based on Performance is inferior to SVM. LIFT is better than LLFN-BSVM, which is because LIFT uses various feature sets to distinguish different labels by performing cluster analysis on positive and negative instances.

According to the analysis above, it is possible to obtain that the LLFN algorithm and LLFN-SVM algorithm are competitive with several other algorithms. A great variety of experimental results show the effectiveness of multi-label learning by jointing neural networks with label-specific features.

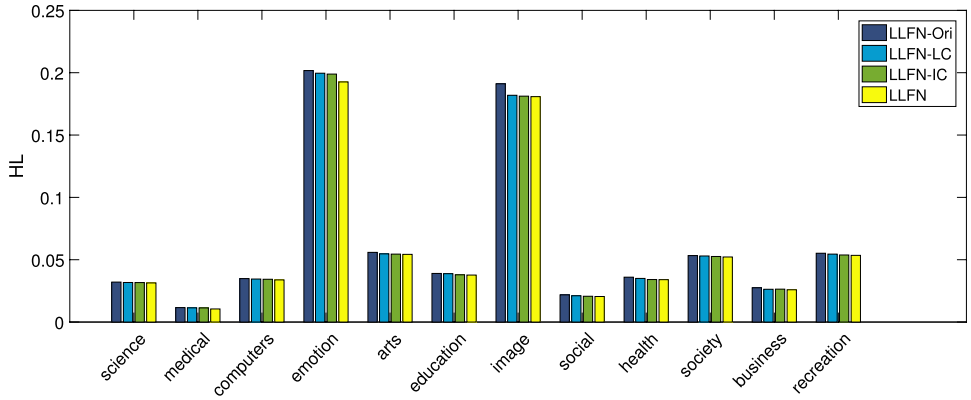
4.5 Component analysis

To further validate the effectiveness of each module of the LLFN algorithm, component analysis experiments were conducted on 12 multi-label datasets , and the experimental results of three evaluation metrics are shown in Fig. 3. Among them, the algorithm LLFN-Ori only considers the extraction of specific features by the neural network and adds l_1-norm without considering any correlation. The algorithm LLFN-LC only adds label correlation, and the algorithm LLFN-IC adds only instance similarity. The algorithm LLFN in this paper adds label correlation and instance correlation at the same time.

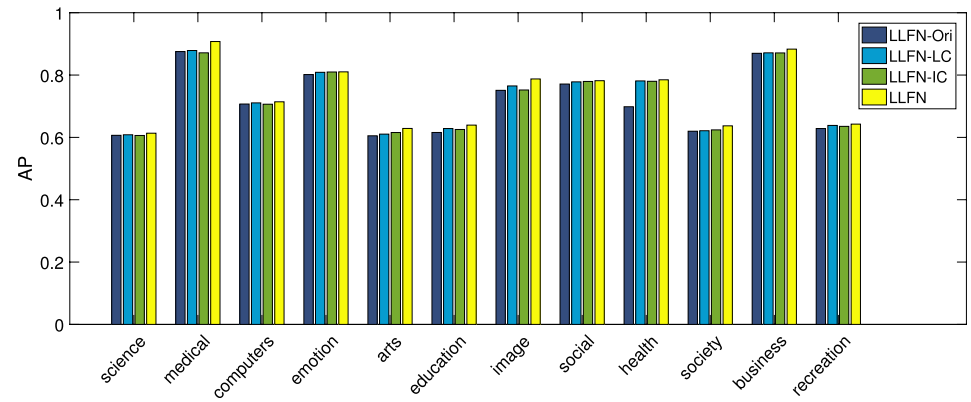
Comparing LLFN-LC, LLFN-IC, and LLFN-Ori, it can be found that LLFN-LC and LLFN-IC outperform LLFN-Ori in all five evaluation metrics on all data sets, which indicates that considering label correlation in label space and instance relevance in feature space alone helps multi-label classification. LLFN is superior to its variant algorithms in most cases. The main reason is that LLFN improves the performance of the algorithm by integrating both label relevance and instance similarity, which confirms the effectiveness of each module of our model.

4.6 Parameter sensitivity analysis

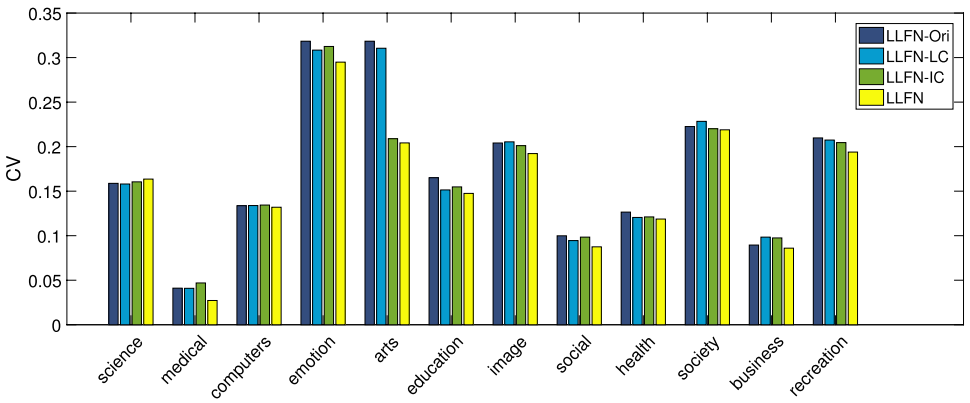
There are 3 basic parameters in the algorithm of this paper α, β, and λ, which respectively control the label correlation, sparsity of label-specific features after input space mapping, and correlation between instances, respectively. In this paper, experiments are conducted on the emotion dataset to investigate the sensitivity of LLFN. As shown in Fig. 4. First, the sensitivity of α and λ is analyzed by fixing an optimal parameter β. We observed that α is almost unchanged when λ changes within $\{2^{-5}, 2^{-4}, \ldots, 2^{1}, 2^{2}\}$, finding that the performance index of LLFN is not sensitive to α, and the



(a) HL↓.



(b) AP↑.

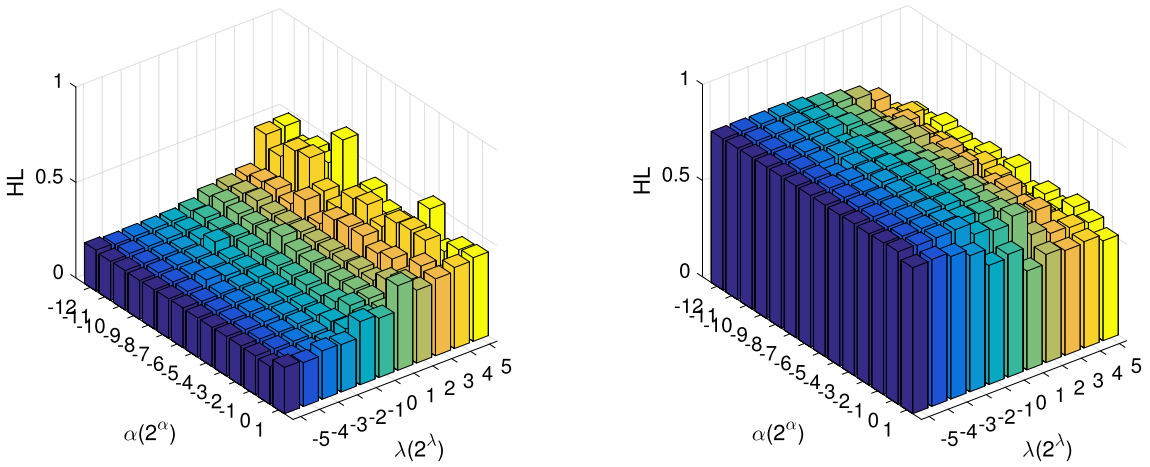


(c) CV↓.

Fig. 3 Five evaluation metrics results of LLFN and its variants on all datasets

best performance is when the value of λ is small. We find an interesting phenomenon that the classification performance gradually decreases with the increase of λ, intuitively because the instance correlation and feature correlation in the real label set is small, and the α peak in the feature space, it means that these two instances can share more label subsets, which affects the experimental results to some extent.

The next step is to study the β effect on the classification performance of the algorithm LLFN by setting the other two parameters to their optimal values: $\alpha = 2^{-1}, \lambda = 2^{-3}$. Figure 5 gives the variation of β under each metric within $\{2^{-6}, 2^{-5}, \ldots, 2^{5}, 2^{6}\}$, and it can be seen that the situation is best when $\beta = 2^{-1}$, and the sparsity constraint for label-specific features cannot be well constrained when β is too small, and the performance drops sharply when β is too large. This is because too large β will cause most elements



(a) HL. **(b)** AP.

Fig. 4 Sensitivity analysis of LLFN under different input values of α and λ

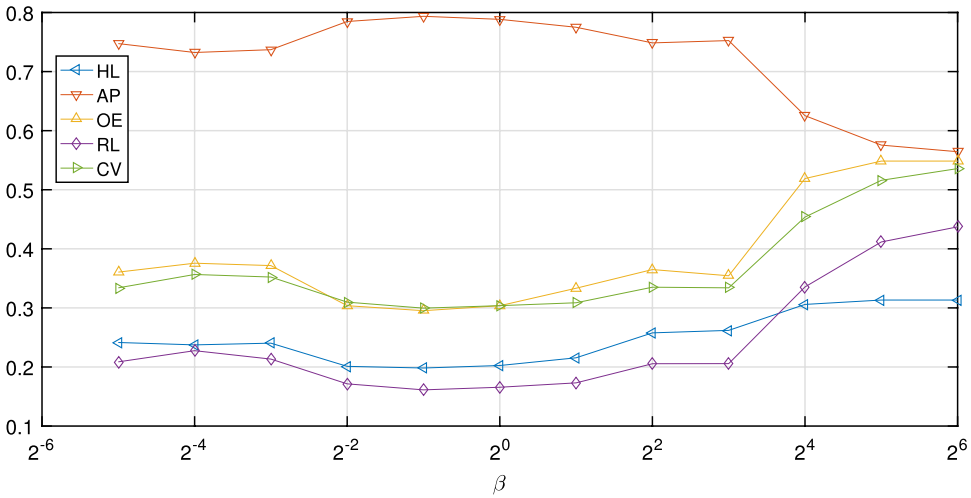
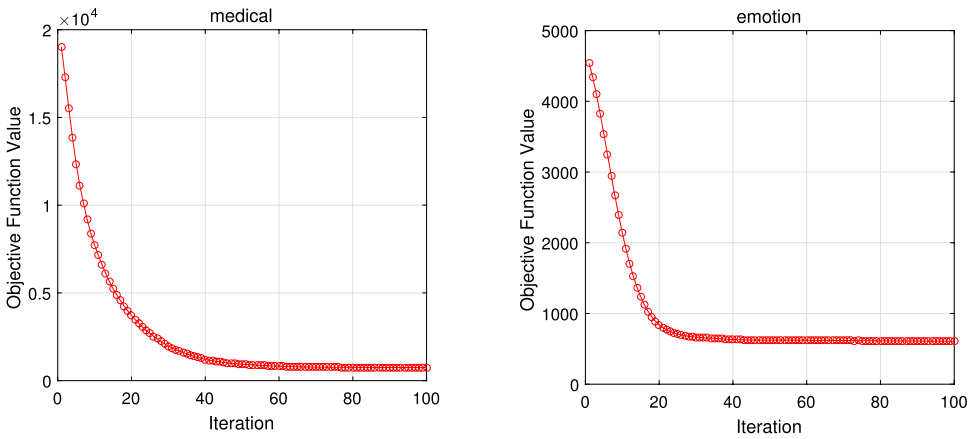


Fig. 5 Sensitivity analysis of β, where $\alpha = 2^{-1}$, $\lambda = 2^{-3}$ and $\beta \in \{2^{-6}, 2^{-5}, \ldots, 2^5, 2^6\}$



(a) medical. **(b)** emotion.

Fig. 6 Convergence trend analysis

of the coefficient matrix $\boldsymbol{W}$ to be zero, and some features will be ignored, which leads to a decrease in classification performance

4.7 Convergence analysis

As mentioned earlier, the proposed algorithm LLFN in this paper solves the optimal solution by the accelerated proximal gradient method (APG), and the convergence rate of the APG for a given appropriate step size is $O(t^{-2})$. Figure 6 shows the number of iterations of the objective loss function of LLFN on the two datasets education and emotion, the objective function decreases sharply and stabilizes after 60 iterations.

5 Conclusion

In this paper, we propose a novel neural network-based multi-label specific feature learning algorithm. Different from many multi-label classification methods, this method learns a low-dimensional mapping representation of the original feature space through a neural network, uses the instance feature space as the input layer of the neural network, and eventually obtains the label space as the output layer after the processing of the hidden layer. Meanwhile, the empirical minimization loss function is used to learn the specific features of the labels. Finally, label correlation and instance similarity are introduced for multi-label classification. The experimental results demonstrate that the proposed algorithm is effective in multi-label classification, and compared with many state-of-the-art algorithms, the proposed algorithm has better performance. However, the results of our proposed algorithm are not very satisfactory when dealing with multi-label datasets with a small amount of samples, which is the part that we will optimize and study subsequently. Currently, the research on multi-label classification is widely used. Next, we expect to extend our proposed algorithm to practical application scenarios for related research. Furthermore, We present experimental results of the algorithm in this paper on SVM classifiers, but we are also interested in extending this technique to other classifiers.

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. Gargiulo F, Silvestri S, Ciampi M, De Pietro G (2019) Deep neural network for hierarchical extreme multi-label text classification. Appl Soft Comput 79:125–138
2. Li Y, Song Y, Luo J (2017) Improving pairwise ranking for multi-label image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3617–3625
3. Wen S, Liu W, Yang Y, Zhou P, Guo Z, Yan Z, Chen Y, Huang T (2020) Multilabel image classification via feature/label co-projection. IEEE Trans Syst Man Cybern Syst 51:7250–7259
4. Gull S, Shamim N, Minhas F (2019) Amap: hierarchical multi-label prediction of biologically active and antimicrobial peptides. Comput Biol Med 107:172–181
5. Liu L, Tang L, Jin X, Zhou W (2019) A multi-label supervised topic model conditioned on arbitrary features for gene function prediction. Genes 10(1):57
6. Zhang M-L, Zhou Z-H (2013) A review on multi-label learning algorithms. IEEE Trans Knowl Data Eng 26(8):1819–1837
7. Boutell MR, Luo J, Shen X, Brown CM (2004) Learning multi-label scene classification. Pattern Recogn 37(9):1757–1771
8. Zhang M-L, Zhou Z-H (2007) Ml-knn: a lazy learning approach to multi-label learning. Pattern Recogn 40(7):2038–2048
9. 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
10. Weng W, Lin Y, Shunxiang W, Li Y, Kang Y (2018) Multi-label learning based on label-specific features and local pairwise label correlation. Neurocomputing 273:385–394
11. Read J, Pfahringer B, Holmes G, Frank E (2011) Classifier chains for multi-label classification. Mach Learn 85(3):333–359
12. Zhao W, Kong S, Bai J, Fink D, Gomes C (2021) Learning high-order label correlation for multi-label classification via attention-based variational autoencoders. arXiv preprint arXiv:2103.06375
13. Guo B, Hou C, Nie F, Yi D (2016) Pervised multi-label dimensionality reduction. In: 2016 IEEE 16th international conference on data mining (ICDM. IEEE), pp 919–924
14. Øyvind MK, Cristina S-R, Maria BF, Robert J (2019) Noisy multi-label semi-supervised dimensionality reduction. Pattern Recogn 90:257–270
15. Zhang M-L, Lei W (2014) Lift: multi-label learning with label-specific features. IEEE Trans Pattern Anal Mach Intell 37(1):107–120
16. 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
17. Huang J, Li G, Huang Q, Xindong W (2017) Joint feature selection and classification for multilabel learning. IEEE Trans Cybern 48(3):876–889
18. 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
19. Bello M, Nápoles G, Sánchez R, Bello R, Vanhoof K (2020) Deep neural network to extract high-level features and labels in multi-label classification problems. Neurocomputing 413:259–270
20. Nam J, Kim J, Mencía EL, Gurevych I, Furnkranz J (2014) Large-scale multi-label text classification–revisiting neural networks. Joint European conference on machine learning and knowledge discovery in databases. Springer, Berlin, pp 437–452
21. Weizhi Liao Yu, Wang YY, Zhang X, Ma P (2020) Improved sequence generation model for multi-label classification via cnn and initialized fully connection. Neurocomputing 382:188–195
22. Wang J, Yang Y, Mao J, Huang Z, Huang C, Xu W (2016) Cnn-rnn: a unified framework for multi-label image classification. In:

Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2285–2294
23. Zhang M-L (2009) M l-rbf: Rbf neural networks for multi-label learning. Neural Process Lett 29(2):61–74
24. Huimin L, Zhang M, Xing X, Li Y, Shen HT (2020) Deep fuzzy hashing network for efficient image retrieval. IEEE Trans Fuzzy Syst 29(1):166–176
25. Xie Y, Zhang J, Xia Y, Shen C (2020) A mutual bootstrapping model for automated skin lesion segmentation and classification. IEEE Trans Med Imaging 39(7):2482–2493
26. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
27. Cheng Y, Zhao D, Wang Y, Pei G (2019) Multi-label learning with kernel extreme learning machine autoencoder. Knowl-Based Syst 178:1–10
28. Parwez MA, Abulaish M et al (2019) Multi-label classification of microblogging texts using convolution neural network. IEEE Access 7:68678–68691
29. Zhu J, Liao S, Lei Z, Li SZ (2017) Multi-label convolutional neural network based pedestrian attribute classification. Image Vis Comput 58:224–229
30. Nam J, Mencía EL, Kim HJ, Fürnkranz J (2017) Maximizing subset accuracy with recurrent neural networks in multi-label classification. In: Proceedings of the 31st international conference on neural information processing systems, pp 5419–5429
31. Chen SF, Chen YC, Yeh CK, Wang YCF (2018) Order-free rnn with visual attention for multi-label classification. In: Thirty-Second AAAI conference on artificial intelligence
32. Rui H, Liuyue K (2021) Local positive and negative label correlation analysis with label awareness for multi-label classification. Int J Mach Learn Cybern 12:1–14
33. Bidgoli AA, Ebrahimpour-komleh H, Rahnamayan S (2021) A novel binary many-objective feature selection algorithm for multi-label data classification. Int J Mach Learn Cybern 12(7):2041–2057
34. 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
35. Zhu W, Li W, Jia X (2020) Multi-label learning with local similarity of samples. In: 2020 International joint conference on neural networks (IJCNN). IEEE, pp 1–8
36. Zhu Y, Kwok JT, Zhou Z-H (2017) Multi-label learning with global and local label correlation. IEEE Trans Knowl Data Eng 30(6):1081–1094
37. Jie B, Zhang D, Cheng B, Shen D, Initiative ADN (2015) Manifold regularized multitask feature learning for multimodality disease classification. Hum Brain Mapp 36(2):489–507
38. Han H, Mengxing Huang Yu, Zhang XY, Feng W (2019) Multi-label learning with label specific features using correlation information. IEEE Access 7:11474–11484
39. Gersho A, Gray RM (2012) Vector quantization and signal compression, vol 159. Springer, Berlin
40. Abdel-Ghaffar KAS (2019) Sets of binary sequences with small total hamming distances. Inf Process Lett 142:27–29
41. Beck A, Teboulle M (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J Imag Sci 2(1):183–202
42. Lin Z, Ganesh A, Wright J, Wu L, Chen M, Ma Y (2009) Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. Coordinated Science Laboratory Report no. UILU-ENG-09-2214, DC-246
43. 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
44. Tan Y, Sun D, Shi Y, Gao L, Gao Q, Lu Y (2021) Bi-directional mapping for multi-label learning of label-specific features. Appl Intell 52:1–20