Knowledge-Based Systems 218 (2021) 106841

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

# Consistency and diversity neural network multi-view multi-label learning

Dawei Zhao [a,b], Qingwei Gao [a,b,*], Yixiang Lu [b], Dong Sun [b], Yusheng Cheng [c]

[a] School of Computer Science and Technology, Anhui University, Hefei 230601, PR China
[b] School of Electrical Engineering and Automation, Anhui University, Hefei 230601, PR China
[c] School of Computer and Information, Anqing Normal University, Anqing 246011, PR China

ARTICLE INFO

*Article history:*
Received 4 September 2020
Received in revised form 20 December 2020
Accepted 2 February 2021
Available online 10 February 2021

*Keywords:*
Multi-view learning
Multi-label learning
Consistency
Diversity
Neural network

ABSTRACT

In multi-view multi-label learning, each object is represented by multiple heterogeneous data and is simultaneously associated with multiple class labels. Previous studies usually use shared subspaces to fuse multi-view representations. However, as the number of views increases, it is more challenging to capture the high-order relationships among multiple views. Therefore, a novel neural network multi-view multi-label learning framework is proposed, which is intended to solve the problem of consistency and diversity among views through a simple and effective method(CDMM). First, we build a separate classifier for each view based on the neural network method of the nonlinear kernel mapping function and require each view to learn a consistent label result. Then, we consider the diversity of individual views while learning a consistent representation among views. For this reason, we combine the Hilbert–Schmidt Independence Criterion with exploring the diversity among different views. Finally, the label correlation factor is in addition to the classification model, and the view contribution factor is added to the prediction model. A large number of comparative experiments with existing state-of-the-art solutions on benchmark multi-view multi-label learning data sets show the effectiveness of this method.



## 1. Introduction

Multi-label(ML) learning solves the problem of label ambiguity by associating a single instance with a set of labels. For example, a picture can be labeled as the "lake", "mountain", and "forest" at the same time, and there may be a strong correlation among these labels. The main task of ML learning is to build a learning model, which can effectively predict the possible label set of unknown objects. In the past few decades, scholars have proposed many effective ML learning algorithms in various domains, such as image annotation [1,2], video annotation [3], and bioinformatics [4,5] etc.

Traditional ML learning is to learn knowledge from a single data structure. However, in the real world, due to the increasing diversity of data collection and feature extraction methods, a single example has multiple views. This type of data is usually associated with numerous heterogeneous feature representations simultaneously, and each feature representation provides a different view of the data [6–9]. For example, in image classification, natural scene images can usually be reproduced by visual features or described by a specific text. The main challenge of this type of task is how to effectively learn the heterogeneity among multiple views while accurately classifying data. Therefore, in the face of more complex data classification problems in real scenes, a multi-view multi-label(MVML) learning framework has emerged.

Due to the widespread existence of MVML datasets, MVML learning has become an active research area in many practical applications [10,11]. In MVML learning, each instance is represented by multiple heterogeneous feature data, which is also associated with multiple class labels. Both solutions based on multi-view(MV) learning or ML learning have their fundamental problems to be resolved. The main issues to be solved urgently in the method that focuses on MV learning are:

1. There should be consistent information representation among different views, so how to effectively solve the consistency problem among mining views and fusing the correlation between high-dimensional heterogeneous data is of paramount importance.
2. Information observed among views is different, so there is individual diversity in the information obtained. Individual diversified information mining contributes to enhancing communication among views, thereby improving the performance of the algorithm.

* Corresponding author at: School of Computer Science and Technology, Anhui University, Hefei 230601, PR China.
*E-mail address:* qingweigao@ahu.edu.cn (Q. Gao).

https://doi.org/10.1016/j.knosys.2021.106841
0950-7051/

3. The structural differences of the data among the various views lead to different importance of each view, so the contribution degree of each view is also different.

Based on these problems, we divide these algorithms into two types. The first type of method is a two-step learning strategy. The first step directly uses the MV learning method to solve the MV learning problem. The second step utilizes the existing ML learning method to solve the ML learning problem. For example, Liu et al. [12] proposed a multi-view framework lrMMC based on matrix decomposition. The framework first seeks the shared representation of multiple views and then completes the classification based on the matrix of the shared feature space. Furthermore, [11] maps each view to a shared space to eliminate noise and redundancy while maintaining the sparse and manifold structure of the image data, respectively. This two-step learning strategy learning often results in sub-optimal results.

The second type of method is the joint learning strategy. They established a unified, MVML learning model to solve the problem. For example, Zhao et al. [13] introduced a predictive reliability measurement method to select samples for sharing labels to share information with other opinions in a co-training manner; Luo et al. [14] jointly extract the consistency and specificity of heterogeneous features for subspace learning; Zhang et al. [15] proposed an MVML method based on matrix factorization, which uses the complementarity among different views to obtain a common semantic representation. The complexity of this type of method is high, but the performance of the model has been significantly improved.

The main problems that need to be solved urgently in the method that focuses on ML learning are:

1. In the face of high-dimensional heterogeneous data, how to effectively predict the label set of unknown instances;
2. Effectively fusion of information among each view and mining label correlation to improve the performance of the classifier.

Based on this, some scholars have proposed solutions. One type of method is tantamount to connect all the data views in series to one data view, and then use the ML learning method to solve it. However, this concatenation strategy has the following problems: it ignores the different physical interpretations of the view data features; the concatenation strategy causes the feature dimension of the data to be too large, and the model training will overfit when the training; the other type establishes an ML classification model for each MV heterogeneous data, and unify the results of these models to construct the final prediction model. For example, Ren et al. [16] fuse multiple views into a mixed feature matrix and use low-rank structure and manifold regularization to utilize global label correlation and local smoothness. Nevertheless, parallel strategy forces all views to output consistent results, ignoring the diversity among different views. Another common problem in MVML learning is that there is an individual difference in the degree of contribution among other views.

Based on the above analysis, it can be seen that the current MVML learning mainly faces the following challenges: (1) the problems of consistency, diversity, and contribution degree among views. (2) the problem of label correlation in ML learning. Some existing learning methods that adopt step-by-step strategies ignore the information exchange between MV and ML when solving MVML problems, so they often get sub-optimal results. However, the learning method that adopts a unified strategy will have excessive model complexity or insufficient consideration of critical issues. To effectively solve the current problems facing MVML learning, a consistency and diversity neural network multi-view multi-label learning method was proposed and named CDMM. First, we design a random single-hidden layer feedforward neural network(SLFN) to perform ML learning for each view to ensure the consistency of all views. Then, we use the Hilbert–Schmidt Independence Criterion(HSIC) [17–19] to induce diversity among different views to learn the diversity information represented by different views. Finally, in the classification, we combine the advantages of the proposed classifier to enrich the original label space with the idea of label dependence propagation; in the prediction, we make the final prediction according to the different effects of each view on the contribution of the MVML learning task.

The main contributions of the paper are as follows:

1. An MVML feedforward neural network model is constructed, different from the traditional neural network model. The CDMM does not require iteration, it is efficient and straightforward, and its parts are closely integrated.
2. CDMM has a unified framework to jointly study the view consistency and diversity issues in MVML learning, and at the same time, integrate the correlation of labels and different contributing factors of views into the classification and prediction models. In addition, a similar ensemble learning method is used to predict the final model.
3. A large number of empirical results of CDMM on the benchmark data set proves that it has certain advantages compared with some related and competitive methods.

The remainder of this paper is organized as follows. In Section 2, the related work of MVML learning is briefly introduced. Section 3 introduces the technical details of CDMM. The results of comparative experiments and specific analyses are illustrated in Section 4. Finally, Section 5 concludes this paper.

## 2. Related work

### 2.1. Multi-label learning

The difference from traditional single-label learning tasks is that the goal of ML learning is to assign multiple class labels for a single instance, which has attracted the attention and research of a large number of scholars in different machine learning tasks. According to different types of label correlations used, the existing ML methods can usually be divided into three categories. First-order: Consider that each label has its unique attributes and ignores the correlation among labels, such as Binary Relevance(BR) [1], ML-*k*NN [20], and LIFT [21]. FRS-LIFT [22] is an extended algorithm on LIFT, which implements label-specific feature reduction through fuzzy rough sets. However, first-order strategies often get sub-optimal results; Second-order: Consider the correlation between paired labels, such as Calibrated Label Ranking(CLR) [23], MLRL [24], but the relationship among labels in the real world is usually more complicated; high-level: Mining the correlation among all category labels. For example, Xu et al. [25] proposed an ensemble framework for training multi-label models while using a low-rank structure to capture the complex correlation among labels. MLMF [26] learns the label correlation and assumes that the learning of each label depends on the approximate output of the feature vector and other labels; LSML [27] improves the performance of multi-label classification by jointly learning label correlation and label-specific features; WML-LSC [28] divides the noisy feature matrix into a feature information-rich matrix and outliers through matrix factorization and uses a linear self-recovery model to reconstruct the original label matrix. It can not only learn robust multi-label models but also refine the noise data of features and labels to a certain extent. In addition, most of the existing ML methods on the hypothesis of label correlation are considered global label correlation. Huang

et al. [29] first proposed the concept of local label correlation when solving ML learning problems.

Although the above methods have reached the most advanced level in processing ML data, they mainly focus on single-view data. There have been some ML learning methods to solve the MV problem. However, they did not explore the heterogeneous relationship among each view, which is of great importance to the successful construction of multi-view learning models [16,30]. Besides, label correlations as an essential influencing factor are also widely used in MVML learning [31,32].

### 2.2. Multi-view multi-label learning

Since a single-view can no longer effectively address the issue of data on diversification, MVML learning has become an active research field in many practical applications [10,11]. In this section, we classify the MVML algorithm from the different view fusion periods: early fusion and late fusion [30].

Early fusion: The typical way is to learn a shared space representation for all heterogeneous feature data, and then construct an ML classification model based on the shared information. For example, Zhu et al. [33] proposed an MVML learning method for incomplete views. First, learn label-specific features through global and local label correlation, then use a low-rank hypothesis matrix to recover incomplete views, and finally use consistent multi-view representations for different complementary information of the view is encoded. LSA-MML [15] solves the MVML learning problem based on the premise that there is a common representation among different views and obtains undiscovered latent semantics through alignment among different views in the kernel space. The goal of MVMLE [8] and LSA-MML is to use the HSIC during the mapping process to maintain consensus on the multi-view potential space. MVD-ELM [34] is a multi-view deep neural network based on extreme learning machine local receptive fields(ELM-LRF) to realize a fast and high-quality projection feature learning method for three-dimensional shapes. It is used to solve the problem of deep learning applications in ML 3D shape segmentation. Tan et al. [35] proposed an individual and commonality-based MVML learning(ICM2L) method to explicitly explore the individuality and commonality information about MVML data in a unified model.

The above method has the following two different problems. (1) This type of method maps each view to a shared subspace to find shared information among all views. However, in this process, there is usually no communication among the various views, and it is challenging to ensure that the shared semantic information among the views is fully utilized. (2) The potential fact that each view has a different specific contribution to ML prediction is ignored.

Therefore, some scholars attempt to solve the above problems. SIMM [36] optimizes a confusing adversarial loss and an ML loss to extract the shared information among views. Secondly, orthogonal constraints are in addition to use the view-private discrimination information, and finally, pass the synergy of shared and private information for semantic learning. TMV-LE [37] explores the high-order relationship among multiple views by constructing the mutual constraints between the tensor factorization and the mapping matrix. Then, the multi-view is utilized for more comprehensively mine the topological structure in the feature space and migrate it to the label space to obtain the label distribution. The key to the effectiveness of the early fusion strategy is how to learn an effective common representation, but it is not easy to learn an accurate, common representation as to the number and types of views increases.

Later fusion: this type of method usually constructs an ML classifier for all views that combine the classification results of each view to make the final label prediction. For example, VLSF [30] learns the label specific representation of each data view, and uses label correlation and view consensus to build an ML classification model. MLSO [38] builds support vector machines(SVM) classifier for each heterogeneous data and jointly learns multi-source ML learning tasks under a unified optimization framework. The problem with the later fusion strategy is that it constructs a separate classification model for each view, which improves the accuracy. However, the complexity of the algorithm also increases accordingly. Besides, the redundancy and the influence of noisy data among the views are not considered.

Based on the above research, we found that the early fusion method has the problem that it is difficult to obtain an accurate, common representation of all views. Moreover, there is no algorithm to simultaneously explore the consistency and complementary knowledge between the views in the later fusion. Based on this, we consider the excellent performance of the neural network method on MVML and propose a novel single hidden layer feedforward neural network MVML algorithm named CDMM. A large number of empirical results on the benchmark data set show that the CDMM method is superior to these related competitive methods.

**Table 1**
Notations and their corresponding definitions.

| Notation | Description |
|---|---|
| $\boldsymbol{X} = \{\boldsymbol{X}^v\}_{v=1}^V$ | Multi-view matrices |
| $\boldsymbol{Y}$ | The label matrix |
| $\boldsymbol{A}$ | Label correlation matrix |
| $\boldsymbol{T}$ | Label completion matrix |
| $\boldsymbol{C}$ | The Laplacian matrix of $\boldsymbol{A}$ |
| $\boldsymbol{H}$ | The hidden layer output matrix |
| $\boldsymbol{\beta}$ | The neural network weight coefficient |
| $\boldsymbol{K}$ | The Gram matrices |
| $\boldsymbol{\theta}^v$ | Contribution weight coefficient of different views |
| $\boldsymbol{N}$ | Number of samples |
| $\boldsymbol{d}$ | Number of features |
| $\boldsymbol{m}$ | Number of labels |
| $\boldsymbol{V}$ | Number of views |

## 3. Proposed approach

### 3.1. Problem statement and notations

Let $\boldsymbol{X} = \left\{\boldsymbol{X}^v\right\}_{v=1}^V$ represents a feature space dataset with $v$ views, where $\boldsymbol{X}^v = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^{\mathrm{T}} \in \mathbb{R}^{N\times d}$ represents the entire feature space of the $v$th view with $N$ samples. $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m] \in \mathbb{R}^{N\times m}$ represents the corresponding label space, where $\boldsymbol{y}_i \in \{-1, 1\}^{N\times m}$ is the label vector of $\boldsymbol{x}_i$, and $m$ represents the number of labels. Table 1 summarizes the definitions of some notations used in this paper.

### 3.2. Label matrix reconstruction

Label correlation is a crucial factor to improve the performance of ML learning, so in this section, we embed the label correlation into the original label matrix. After reconstructing the original label matrix, the member labels in the enhanced label matrix have strong interdependence and weak dependence on other non-member labels. Specifically, inspired by the literature [39], we use two different sources when constructing the label correlation matrix $\boldsymbol{A}$: label space $\boldsymbol{Y}$ and instance space $\boldsymbol{X}$. In the label space, we use *Jaccard* similarity to measure the dependency between two labels:

$$\boldsymbol{A}_{jk}^{(L)} = \frac{\sum_{i=1}^N \boldsymbol{y}_{ij} \cdot \boldsymbol{y}_{ik}}{\sum_{i=1}^N \left(\boldsymbol{y}_{ij} + \boldsymbol{y}_{ik} - \boldsymbol{y}_{ij} \cdot \boldsymbol{y}_{ik}\right)} \tag{1}$$

In the instance space, to model the affinity matrix $A_{jk}^{v(I)}$ for each view, we define:

$$\boldsymbol{A}_{jk}^{v(I)} = \begin{cases} \exp\left(-\frac{\left\|\mu_j^v - \mu_k^v\right\|_2^2}{\sigma^2}\right) & if \left|\boldsymbol{A}_{jk}^{v(I)}\right| \geqslant \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\mu_j^v = \frac{\sum_{i=1}^N x_i^v \cdot y_{ij}}{\sum_{i=1}^N y_{ij}}$. Furthermore, the parameter $\sigma$ in Eq. (2) is fixed to be 1. We combine the above two measurement methods into a weight matrix $\boldsymbol{A}$ in the following way:

$$\boldsymbol{A}_{jk} = \begin{cases} \eta \boldsymbol{A}_{jk}^{(L)} + (1-\eta)\frac{1}{V}\sum_{v=1}^{V} \boldsymbol{A}_{jk}^{(I)} & \left(\boldsymbol{A}_{jk} > \varepsilon\right) \\ 0 & \left(\boldsymbol{A}_{jk} \leqslant \varepsilon\right) \end{cases} \tag{3}$$

Finally, inspired by the idea of label propagation dependence [40], we have:

$$\boldsymbol{T} = \boldsymbol{YC} \tag{4}$$

where $\eta \in [0, 1]$ is a balance factor, $\boldsymbol{C} = \boldsymbol{D} - \boldsymbol{A}$ is the Laplacian matrix of $\boldsymbol{A}$, $\boldsymbol{D}$ is the diagonal matrix $\boldsymbol{D} = \left(\boldsymbol{D}_{ij}\right) = \left(\sum_k \boldsymbol{A}_{jk}\right)$, $\boldsymbol{T} = [\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_m] \in \mathbb{R}^{N\times m}$.

### 3.3. Problem formulation

Different from the widely adopted multi-layer network embedding structure [41], we use a random SLFN method to construct the initial model, similar to the extreme learning machines(ELM) [42], Schmidt et al. [43], and random vector connection function (Random Vector Functional-Link, RVFL) [44] input the weight parameters of the neural network randomly. The details are as following:

$$\sum_{i=1}^{L} g_i\left(\boldsymbol{x}_j\right)^{\mathrm{T}} \boldsymbol{\beta}_i = \sum_{i=1}^{L} g\left(\boldsymbol{w}_i \boldsymbol{x}_j + \boldsymbol{b}_i\right)^{\mathrm{T}} \boldsymbol{\beta}_i = \boldsymbol{o}_j \quad j = 1, \ldots, N \tag{5}$$

where $\boldsymbol{x}_j = \left[\boldsymbol{x}_{j1}, \boldsymbol{x}_{j2}, \ldots, \boldsymbol{x}_{jd}\right]^{\mathrm{T}} \in \mathbb{R}^{d\times N}$ is the input instance feature vector, $\boldsymbol{w}_i = [\boldsymbol{w}_{1i}, \boldsymbol{w}_{2i}, \ldots, \boldsymbol{w}_{di}]^{\mathrm{T}}$ is the weight vector that connects the node of the input feature layer with the node of the $i$th hidden layer, $\boldsymbol{\beta}_i = \left[\boldsymbol{\beta}_{1i}, \boldsymbol{\beta}_{2i}, \ldots, \boldsymbol{\beta}_{mi}\right]^{\mathrm{T}}$ is the weight vector connecting the $i$th hidden node and the output node, and $\boldsymbol{b}_i$ is the bias of the $i$th hidden node. The standard SLFN of $L$ hidden nodes with activation function $g$ can approximate these $N$ samples with zero error, which means that $\sum_{j=1}^{N} \left\|\boldsymbol{o}_j - \boldsymbol{t}_j\right\| = 0$, that is, there are $\boldsymbol{\beta}_i$, $\boldsymbol{w}_i$, and $\boldsymbol{b}_i$:

$$\sum_{i=1}^{L} \boldsymbol{g}_i\left(\boldsymbol{x}_j\right)^{\mathrm{T}} \boldsymbol{\beta}_i = \sum_{i=1}^{L} g\left(\boldsymbol{w}_i \boldsymbol{x}_j + \boldsymbol{b}_i\right)^{\mathrm{T}} \boldsymbol{\beta}_i = \boldsymbol{t}_j \quad j = 1, \ldots, N \tag{6}$$

Eq. (6) can be written succinctly as:

$$\boldsymbol{H}\boldsymbol{\beta} = \boldsymbol{T}$$
$$\boldsymbol{H} = \begin{bmatrix} g\left(\boldsymbol{w}_1 \boldsymbol{x}_1 + \boldsymbol{b}_1\right)^{\mathrm{T}} & \cdots & g\left(\boldsymbol{w}_L \boldsymbol{x}_1 + \boldsymbol{b}_L\right)^{\mathrm{T}} \\ \vdots & \ddots & \vdots \\ g\left(\boldsymbol{w}_1 \boldsymbol{x}_N + \boldsymbol{b}_1\right)^{\mathrm{T}} & \cdots & g\left(\boldsymbol{w}_L \boldsymbol{x}_N + \boldsymbol{b}_L\right)^{\mathrm{T}} \end{bmatrix}_{N\times L} \tag{7}$$

where $\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{\beta}_L^{\mathrm{T}} \end{bmatrix}_{L\times m}$, and $\boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{t}_N^{\mathrm{T}} \end{bmatrix}_{N\times m}$. $\boldsymbol{H}$ is called the hidden layer output matrix of the neural network:

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{H}\boldsymbol{\beta} - \boldsymbol{T}\|_F^2 \tag{8}$$

Further extending the network to MVML data includes:

$$\min_{\boldsymbol{\beta}} \sum_{v=1}^{V} \left(\left\|\boldsymbol{H}^v \boldsymbol{\beta}^v - \boldsymbol{T}\right\|_F^2\right);$$
$$\boldsymbol{H}^v = \begin{bmatrix} g\left(\boldsymbol{w}_1^v \boldsymbol{x}_1^v + \boldsymbol{b}_1^v\right)^{\mathrm{T}} & \cdots & g\left(\boldsymbol{w}_L^v x_1^v + \boldsymbol{b}_L^v\right)^{\mathrm{T}} \\ \vdots & \ddots & \vdots \\ g\left(\boldsymbol{w}_1^v \boldsymbol{x}_N^v + \boldsymbol{b}_1^v\right)^{\mathrm{T}} & \cdots & g\left(\boldsymbol{w}_L^v \boldsymbol{x}_N^v + \boldsymbol{b}_L^v\right)^{\mathrm{T}} \end{bmatrix}_{N\times L} \tag{9}$$

where $v = 1, \ldots, V$, $V$ represents the number of views. Different from the traditional unsupervised multi-view view-consistency learning, a supervised learning model is adopted, which uses consistent information among different views by requiring different views to predict the same label result.

### 3.4. View diversity learning

In this section, we will enhance the diversity of all views to explore the view-diversity information among different views. We consider that higher independence means the higher diversity between the two variables, the lower correlation between the variables at this time to approximate the quantification of diversity based on the dependence among these variables. Here, the HSIC a simple and solid theoretical foundation and the ability to measure linear and nonlinear dependence among variables so that we use it. HSIC estimates the correlation of calculating the square norm of the cross-covariance operator on $\boldsymbol{H}^v\boldsymbol{\beta}^v$ and $\boldsymbol{H}^s\boldsymbol{\beta}^s$ in Hilbert space, and it is given by:

$$HSIC\left(\boldsymbol{H}^v\boldsymbol{\beta}^v, \boldsymbol{H}^s\boldsymbol{\beta}^s\right) = (n-1)^{-2} tr\left(\boldsymbol{K}^v \boldsymbol{P} \boldsymbol{K}^s \boldsymbol{P}\right) \tag{10}$$

In this paper, we use the inner product kernel to specify $\boldsymbol{K}^s = \boldsymbol{H}^s\boldsymbol{H}^{s\mathrm{T}}$. Then minimize the overall HISC on the $v$th individuality matrix to reduce the redundancy between them. The formula is as following:

$$\begin{aligned} \Phi\left(\left\{\boldsymbol{H}^v\boldsymbol{\beta}^v\right\}_{v=1}^V\right) &= \sum_{v=1, v\neq s}^{V} HSIC\left(\boldsymbol{H}^v\boldsymbol{\beta}^v, \boldsymbol{H}^s\boldsymbol{\beta}^s\right) \\ &= \sum_{v=1, v\neq s}^{V} (n-1)^{-2} tr\left(\left(\boldsymbol{H}^v\boldsymbol{\beta}^v\right)^{\mathrm{T}} \boldsymbol{P}\boldsymbol{K}^s\boldsymbol{P}\boldsymbol{H}^v\boldsymbol{\beta}^v\right) \\ &= \sum_{v=1}^{V} tr\left(\left(\boldsymbol{H}^v\boldsymbol{\beta}^v\right)^{\mathrm{T}} \widetilde{\boldsymbol{K}}^v \boldsymbol{H}^v\boldsymbol{\beta}^v\right) \end{aligned} \tag{11}$$

where $\widetilde{\boldsymbol{K}}^v = (n-1)^{-2}\sum_{s=1, s\neq v}^{V} \boldsymbol{P}\boldsymbol{K}^s\boldsymbol{P}$. $\boldsymbol{K}^v$, $\boldsymbol{K}^s$, $\boldsymbol{P} \in \mathbb{R}^{N\times N}$, $\boldsymbol{K}^v$ and $\boldsymbol{K}^s$ are used to measure the kernel-induced similarity between $\boldsymbol{H}^v\boldsymbol{\beta}^v$ and $\boldsymbol{H}^s\boldsymbol{\beta}^s$ vectors, respectively. $\boldsymbol{P} = \boldsymbol{\delta}_{ij} - 1/N$, $\boldsymbol{\delta}_{ij} = 1$, if $i = j$, $\boldsymbol{\delta}_{ij} = 0$ otherwise. Adding Eq. (9) to Eq. (11), the final optimization model is as following:

$$\min_{\boldsymbol{\beta}} \sum_{v=1}^{V} \left(\left\|\boldsymbol{H}^v\boldsymbol{\beta}^v - \boldsymbol{T}\right\|_F^2 + \alpha tr\left(\left(\boldsymbol{H}^v\boldsymbol{\beta}^v\right)^{\mathrm{T}} \widetilde{\boldsymbol{K}}^v \boldsymbol{H}^v\boldsymbol{\beta}^v\right)\right) \tag{12}$$

where $\alpha$ represents a non-negative trade-off parameter, which controls the diversity among views.

### 3.5. Model optimization and prediction

Different from the traditional SLN method which uses iterative backpropagation to solve the weight, we adopt the same scheme as the standard ELM [42,45], replacing backpropagation with a one-time matrix inverse operation and canceling the iterative operation of the neural network. Specifically, the derivation of Eq. (12) w.r.t $\boldsymbol{\beta}$ is as following:

$$\boldsymbol{\beta}^v = (\boldsymbol{H}^v)^{\dagger}\boldsymbol{T} \tag{13}$$

$$\begin{cases} (\boldsymbol{H}^v)^\dagger = \left((\boldsymbol{H}^v)^{\mathrm{T}}\boldsymbol{H}^v + (\boldsymbol{H}^v)^{\mathrm{T}}\widetilde{\boldsymbol{K}}^v\boldsymbol{H}^v\right)^{-1}(\boldsymbol{H}^v)^{\mathrm{T}} \\ (\boldsymbol{H}^v)^\dagger = (\boldsymbol{H}^v)^{\mathrm{T}}\left(\boldsymbol{H}^v(\boldsymbol{H}^v)^{\mathrm{T}} + \widetilde{\boldsymbol{K}}^v\boldsymbol{H}^v(\boldsymbol{H}^v)^{\mathrm{T}}\right)^{-1} \end{cases} \tag{14}$$

When $N \ll L$, then $(\boldsymbol{H}^v)^\dagger$ is the Moore–Penrose generalized inverse of matrix $\boldsymbol{H}$, and $\boldsymbol{H}$ is the hidden layer output matrix. Besides, according to the ridge regression theorem, it is recommended to add a positive value $1/E$ to the diagonal of $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}$ or $\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$ to calculate the output weight [46], which can get a more stable solution and has better generalization performance. In other words, to improve the stability of the Eq. (14), we can have:

$$\boldsymbol{\beta}^v = (\boldsymbol{H}^v)^{\mathrm{T}}\left(\boldsymbol{H}^v(\boldsymbol{H}^v)^{\mathrm{T}} + \widetilde{\boldsymbol{K}}^v\boldsymbol{H}^v(\boldsymbol{H}^v)^{\mathrm{T}} + \frac{\boldsymbol{I}}{E}\right)^{-1}\boldsymbol{T} \tag{15}$$

According to Eq. (15), our final output function can be expressed as:

$$\begin{aligned} &f(\boldsymbol{x}^v) = h(\boldsymbol{x}^v)\boldsymbol{\beta} \\ &= h(\boldsymbol{x}^v)(\boldsymbol{H}^v)^{\mathrm{T}}\left(\left(\boldsymbol{I} + \widetilde{\boldsymbol{K}}^v\right)\boldsymbol{H}^v(\boldsymbol{H}^v)^{\mathrm{T}} + \frac{\boldsymbol{I}}{E}\right)^{-1} \end{aligned} \tag{16}$$

Besides, when the hidden layer feature map $h(\boldsymbol{x})$ is unknown, a kernel matrix can be defined by Eq. (16) (in this paper, we select the Radial Basis Function(RBF) kernel):

$$\boldsymbol{\Omega}_{\mathrm{ELM}} = \boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} : \boldsymbol{\Omega}_{ij} = h(\boldsymbol{x}_j)\cdot h(\boldsymbol{x}_i) = \mathrm{Kernel}(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{17}$$

At this time, the number of hidden nodes $L$ (the dimension of the hidden layer feature space) does not need to be specified. Eq. (16) can be abbreviated as:

$$\begin{aligned} &f(\boldsymbol{x}^v) = h(\boldsymbol{x}^v)\boldsymbol{\beta} \\ &= h(\boldsymbol{x}^v)(\boldsymbol{H}^v)^{\mathrm{T}}\left((\boldsymbol{I} + \widetilde{\boldsymbol{K}}^v)\boldsymbol{H}^v(\boldsymbol{H}^v)^{\mathrm{T}} + \tfrac{\boldsymbol{I}}{E}\right)^{-1} \\ &= \begin{bmatrix} \mathrm{Kernel}(\boldsymbol{x}^v, \boldsymbol{x}_1^v) \\ \vdots \\ \mathrm{Kernel}(\boldsymbol{x}^v, \boldsymbol{x}_N^v) \end{bmatrix}\left((\boldsymbol{I} + \widetilde{\boldsymbol{K}}^v)\boldsymbol{\Omega}_{\mathrm{ELM}} + \tfrac{\boldsymbol{I}}{E}\right)^{-1}\boldsymbol{T} \end{aligned} \tag{18}$$

Considered the different contributions of each view in MVML learning, when making predictions, the output of other views should be weighed, as showed below:

$$\begin{aligned} &\min_{\boldsymbol{\theta}^v} \frac{1}{2}\sum_{v=1}^{V}\boldsymbol{\theta}^v\boldsymbol{PreLoss}^v + \frac{\lambda}{2}\left\|\boldsymbol{\theta}^v\right\|_2^2 \\ &\text{s.t.}\boldsymbol{\theta}^v > 0, \sum_{v=1}^{V}\boldsymbol{\theta}^v = 1 \end{aligned} \tag{19}$$

Solve the objective function Eq. (19) by Lagrangian multiplier method:

$$\boldsymbol{\theta}^v = \begin{cases} \frac{\sum_{v=1}^{V}\boldsymbol{PreLoss}^v + \lambda - V\boldsymbol{PreLoss}^v}{V\lambda} & if \sum_{v=1}^{V}\boldsymbol{PreLoss}^v + \lambda > V\boldsymbol{PreLoss}^v \\ \varepsilon & if \sum_{v=1}^{V}\boldsymbol{PreLoss}^v + \lambda \leqslant V\boldsymbol{PreLoss}^v \end{cases} \tag{20}$$

where $\boldsymbol{PreLoss}^v = \left\|\boldsymbol{H}_{train}^v\boldsymbol{\beta}^v - \boldsymbol{T}_{train}\right\|_F^2$, and $\varepsilon$ is a very small non-negative value.

Finally, our prediction function is expressed as:

$$\boldsymbol{T}_{pre} = \sum_{v=1}^{V}\boldsymbol{\theta}^v\boldsymbol{H}_{test}^v\boldsymbol{\beta}^v \tag{21}$$

Based on the above optimization and prediction of the model, the main process of CDMM is summarized in Algorithm 1.

**Algorithm 1** **C**onsistency and **D**ifference Neural Network **M**ulti-view **M**ulti-label Learning(CDMM).

**Input:**
The training data set:$\boldsymbol{X}^v \in \mathbb{R}^{N\times d}$;
The label data set: $\boldsymbol{Y} \in \mathbb{R}^{N\times m}$;
The non-negative trade-off parameters: $\alpha$, $\lambda$, $C$, $\sigma$, and $\eta$;
**Output:**
Final prediction objective function: $\boldsymbol{T}_{pre}$;
Calculate the label correlation matrix by Eq. (3)
Reconstruction of the original label matrix by Eq. (4)
**for** Train Data $v = 1$ to $V$ **do**
  Calculate the kernel matrix $\boldsymbol{\Omega}_{ELM}$ by Eq. (17);
  Calculate neural network output weights by Eq. (15);
  Calculate the prediction function of the training set by Eq. (18);
  Calculate view contribution weight $\boldsymbol{\theta}^v$ by Eq. (20);
**end for**
Calculate the prediction function of the test set by Eq. (21)
**return** $\boldsymbol{T}_{pre}$.

### 3.6. Complexity analysis

Specifically, the complexity of the HSIC function is $\mathcal{O}(N^3)$, but because the matrix $\boldsymbol{P}$ is a sparse matrix, the overall complexity will be lower than $\mathcal{O}(N^3)$. The complexity of solving weight $\boldsymbol{\beta}$ is $\mathcal{O}(N^2m + N^2d_{\max})$. Since $N \gg m$ and $N \gg d_{\max}$, the overall time complexity of CDMM is $\mathcal{O}(N^3)$. In practice, the time cost mainly comes from the size of the sample size.

## 4. Experiments

### 4.1. Datasets

To verify the effectiveness of the CDMM algorithm, we use a total of 7 benchmark MVML data sets for performance evaluation, which can be download from MULAN[1] and LEAR.[2] The details of the data sets are summarized in Table 2.

### 4.2. Comparing algorithms

To verify the performance of CDMM, we selected three state-of-the-art MVML learning algorithms and one incomplete view MVML weak-label learning algorithm as the comparison algorithm.

1. iMvWL[3] [32]: Learning incomplete views and weak labels by sharing subspace information. In the experiment, the complete view information can be obtained. The parameters $\alpha$ and $\beta$ are both searched in $\{10^{-5}, 10^{-4}, \ldots, 10^{0}\}$.

2. ICM2L[4]: Explicitly explore the individuality and common information of MVML data in the unified model. Set $\alpha$, $\beta$ and $k$ within the range suggested in the literature [35].

3. VLSF[5]: Multi-View Multi-Label Learning With View-Label-Specific Features. All parameters are searched within the range given in the literature [30].

1 data sets: http://mulan.sourceforge.net/datasets-mlc.html.
2 data sets: http://lear.inrialpes.fr/people/guillaumin/data.php.
3 code: http://mlda.swu.edu.cn/codes.php?name=iMvWL.
4 code:http://mlda.swu.edu.cn/codes.php?name=ICM2L.
5 code: http://www.escience.cn/people/huangjun/index.html.

**Table 2**
MVML data sets.

| Views | Emotions | Yeast | Pascal07 | Corel5k | Espgame | Iaprtc12 | mirflickr |
|---|---|---|---|---|---|---|---|
| 1 | rhythmic attributes | Genetic Expression | DenseSift | DenseHue | DenseHue | DenseHue | DenseHue |
| | (8) | (79) | (1000) | (100) | (100) | (100) | (100) |
| 2 | timbre attributes | phylogenetic profile | HarrisSift | DenseSift | DenseSift | DenseSift | DenseSift |
| | (64) | (24) | (1000) | (1000) | (1000) | (1000) | (1000) |
| 3 | – | – | Gist(512) | Gist(512) | Gist(512) | Gist(512) | Gist(512) |
| 4 | – | – | HSV(4096) | HSV(4096) | HSV(4096) | HSV(4096) | HSV(4096) |
| 5 | – | – | RGB(4096) | Lab(4096) | Lab(4096) | Lab(4096) | Lab(4096) |
| 6 | – | – | Tags(804) | RGB(4096) | RGB(4096) | RGB(4096) | RGB(4096) |
| Domain | music | biology | image | image | image | image | image |
| $m$ | 6 | 14 | 20 | 260 | 268 | 291 | 457 |
| $N$ | 593 | 2417 | 9963 | 4999 | 20770 | 19627 | 25000 |

4. SIMM[6]: Multi-View Multi-Label Learning with View-Specific Information Extraction. The number of hidden layers is fixed to 64. The parameter $\alpha$ is set to 1, and $\beta$ is searched in $\{0.1, 0.01, 0.001, 0.0001\}$.

5. CDMM: The parameter $\alpha$ is searched in $\{10^{-10}, 10^{-9}, \ldots, 10^{-5}\}$, $\lambda$ is searched in $\{10^{1}, 10^{2}, \ldots, 10^{7}\}$, the regularization coefficient $C$ is searched in $\{10^{-5}, 10^{-1}, \ldots, 10^{5}\}$, the kernel parameter $\sigma$ is searched in $\{10^{-2}, 10^{-1}, \ldots, 10^{2}\}$, and the label correlation balance parameter $\eta$ is explored within the range $\{0.5 - 0.8\}$.

### 4.3. The experimental environment and evaluation metrics

In this paper, the experiment was performed on Windows 10, Intel(R) Core(TM) i7-7700K, and 32GB RAM, and the method was implemented in MATLAB 2016b. We used (1) Hamming Loss(HL), (2) Average Precision(AP), (3) One Error(OE), (4) Ranking Loss(RL), (5) Coverage(CV), (6) Macro-F1, (7) Micro-F1, and (8) Subset Accuracy(SA), eight widely used ML metrics for performance evaluation. They evaluate the performance of the algorithm from the perspective of ranking and classification, respectively. The specific measurement definition can be found in [47]. For Average Precision, Macro-F1, Micro-F1, and Subset Accuracy, the larger the value, the better the performance. For the other four metrics, the smaller the value, the better the performance.

### 4.4. Experimental results and analysis

For each MVML data set, we randomly select 80% of the total data as the training set, and the remaining data as the test set, and repeat the experiment for all comparison algorithms five times. Tables 3, to 4 respectively list the average results (mean $\pm$ std) of the 5 comparison algorithms on the 7 MVML benchmark data sets, and the best results are shown in black. $\downarrow$ means that the smaller the evaluation metrics value is, the better, and $\uparrow$ means that the larger the evaluation metrics value is, the better.

Furthermore, a statistical hypothesis test was utilized to verify and compare the relative performance of various algorithms. The Friedman test [48] was utilized for performance analysis. Table 5 summarizes the Friedman statistics $F_F$ and the corresponding critical values of the various evaluation metrics. As showed in Table 5, at the significance level $\rho = 0.05$, each evaluation metric is rejected when the null hypothesis that all comparison algorithms are equivalently executed. Therefore, the *Nemenyi* test [48] is utilized as a post-hoc test to compare the performance of each algorithm and observe whether the CDMM algorithm is competitive. There is a significant difference in performance between the two classifiers if the corresponding average ranking reaches at least a critical difference (CD):

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

Analysis of experimental results: according to the experimental results reported in Tables 3 and 4, the following observations can be drawn:

- Among 56 configurations (7 data sets and 8 evaluation metrics), CDMM ranks first and second at 64.3% and 26.9%, respectively.
- It is worth noting that CDMM outperforms all comparison algorithms on all metrics on *Emotions* and *Yeast*.
- Except for Pascal07 and Mirflickr, CDMM achieved the best performance on AP, OE, Macro-F1, Micro-F1 and Subset Accuracy.
- All data sets: the HL value of CDMM is almost the same as the optimal result, and the CV and RL are also not much different from the optimal result.

For Nemenyi test, $q = 2.728$ at significance level $\rho = 0.05$, and thus $\mathrm{CD} = 2.31(k = 5, N = 7)$. Fig. 1 indicates the CD diagrams of each algorithm under different evaluation metrics, respectively. In each subfigure, two or more algorithms are connected by solid red lines indicate that there is no significant difference in performance among them. For each approach, there are 40 comparative results (5 parallel approaches and 8 evaluation metrics).

Observing Fig. 1, we can see that under the confidence interval of significance level $\rho = 0.05$, CDMM has a significant advantage over other comparison algorithms in 30% of the cases, ranking first. And SIMM ranked second with 27.5%.

Through the above experimental results, it can be analyzed:

- SIMM, iMvWL, and ICM2L are all designed to use subspace learning to solve the problem of MVML learning, but the performance of the SIMM is always better than the other two. The reason is that SIMM builds a model through a neural network, and while mining the shared subspace, it also extracts the private information of the view.
- The VLSF algorithm performs better than iMvWL and ICM2L, because it pays more attention to the consensus learning among views, and at the same time mines the view-label-Specific to improve the effectiveness of the algorithm. The performance of VLSF is inferior to CDMM mainly because VLSF ignores the diversity of views, and there are limitations in extracting view information.

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

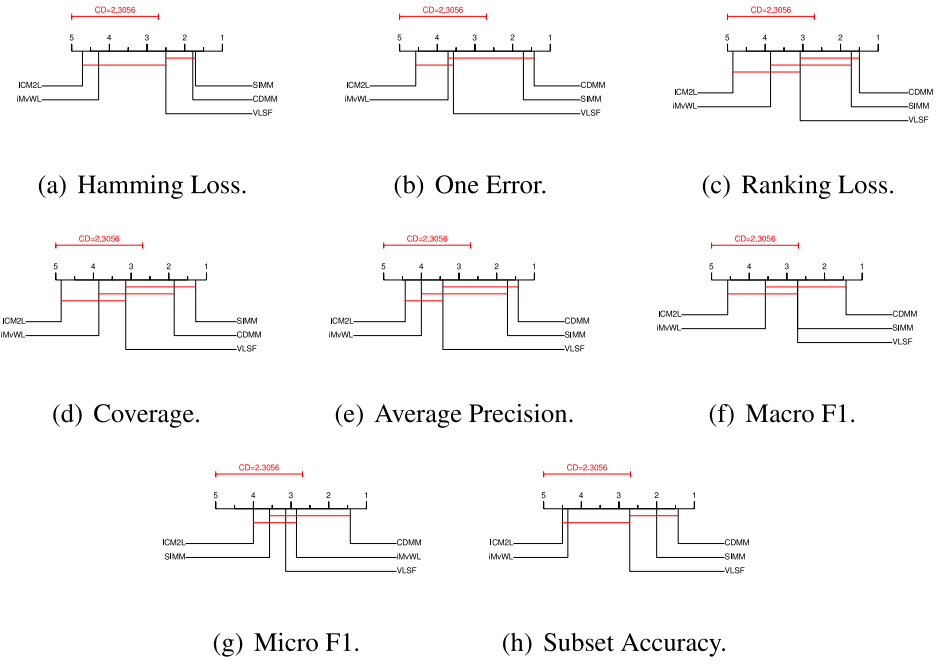
**Table 3**
The results of each comparison algorithm on HL, OE, RL, and CV evaluation metrics (↓) on 7 data sets.

| Dataset | metric | SIMM | ICML | iMvWL | VLSF | CDMM |
|---|---|---|---|---|---|---|
| Emotions | HL | 0.246 ± 0.008 | 0.375 ± 0.015 | 0.395 ± 0.011 | 0.293 ± 0.007 | **0.207 ± 0.014** |
| | OE | 0.310 ± 0.056 | 0.530 ± 0.030 | 0.521 ± 0.021 | 0.539 ± 0.073 | **0.304 ± 0.034** |
| | RL | 0.178 ± 0.024 | 0.443 ± 0.013 | 0.414 ± 0.012 | 0.330 ± 0.022 | **0.174 ± 0.013** |
| | CV | 0.307 ± 0.014 | 0.530 ± 0.036 | 0.506 ± 0.014 | 0.441 ± 0.022 | **0.304 ± 0.018** |
| Yeast | HL | 0.207 ± 0.005 | 0.278 ± 0.008 | 0.269 ± 0.005 | 0.258 ± 0.004 | **0.189 ± 0.006** |
| | OE | 0.225 ± 0.028 | 0.235 ± 0.024 | 0.292 ± 0.020 | 0.343 ± 0.015 | **0.211 ± 0.013** |
| | RL | 0.165 ± 0.008 | 0.215 ± 0.011 | 0.214 ± 0.008 | 0.320 ± 0.006 | **0.151 ± 0.008** |
| | CV | 0.450 ± 0.004 | 0.503 ± 0.006 | 0.494 ± 0.009 | 0.601 ± 0.008 | **0.426 ± 0.008** |
| Pascal07 | HL | **0.046 ± 0.001** | 0.115 ± 0.003 | 0.086 ± 0.002 | 0.050 ± 0.000 | 0.049 ± 0.001 |
| | OE | **0.255 ± 0.008** | 0.589 ± 0.002 | 0.397 ± 0.021 | 0.290 ± 0.014 | 0.308 ± 0.011 |
| | RL | **0.066 ± 0.002** | 0.241 ± 0.040 | 0.138 ± 0.011 | 0.070 ± 0.002 | 0.070 ± 0.002 |
| | CV | **0.106 ± 0.002** | 0.308 ± 0.048 | 0.189 ± 0.015 | 0.110 ± 0.001 | 0.111 ± 0.003 |
| Corel5k | HL | **0.011 ± 0.000** | 0.022 ± 0.000 | 0.022 ± 0.000 | 0.013 ± 0.000 | **0.011 ± 0.000** |
| | OE | 0.363 ± 0.011 | 0.697 ± 0.007 | 0.687 ± 0.003 | 0.438 ± 0.017 | **0.362 ± 0.006** |
| | RL | **0.059 ± 0.002** | 0.149 ± 0.002 | 0.130 ± 0.003 | 0.076 ± 0.004 | 0.069 ± 0.005 |
| | CV | **0.148 ± 0.006** | 0.334 ± 0.000 | 0.286 ± 0.004 | 0.187 ± 0.009 | 0.179 ± 0.011 |
| ESPgame | HL | **0.017 ± 0.000** | 0.029 ± 0.000 | 0.028 ± 0.000 | **0.017 ± 0.000** | 0.018 ± 0.000 |
| | OE | 0.476 ± 0.022 | 0.713 ± 0.030 | 0.674 ± 0.000 | 0.533 ± 0.011 | **0.465 ± 0.009** |
| | RL | **0.120 ± 0.006** | 0.203 ± 0.002 | 0.190 ± 0.002 | 0.142 ± 0.003 | 0.124 ± 0.002 |
| | CV | **0.308 ± 0.012** | 0.479 ± 0.001 | 0.447 ± 0.004 | 0.365 ± 0.005 | 0.337 ± 0.005 |
| Iaprtc12 | HL | **0.019 ± 0.000** | 0.032 ± 0.000 | 0.031 ± 0.000 | **0.019 ± 0.000** | **0.019 ± 0.000** |
| | OE | 0.447 ± 0.021 | 0.720 ± 0.005 | 0.624 ± 0.002 | 0.465 ± 0.006 | **0.439 ± 0.007** |
| | RL | 0.089 ± 0.007 | 0.189 ± 0.001 | 0.165 ± 0.002 | 0.106 ± 0.001 | **0.089 ± 0.002** |
| | CV | **0.270 ± 0.019** | 0.497 ± 0.001 | 0.435 ± 0.003 | 0.327 ± 0.002 | 0.284 ± 0.006 |
| Mirflickr | HL | **0.006 ± 0.000** | 0.013 ± 0.000 | 0.013 ± 0.000 | **0.006 ± 0.000** | **0.006 ± 0.000** |
| | OE | **0.865 ± 0.006** | 0.908 ± 0.004 | 0.887 ± 0.003 | 0.891 ± 0.003 | 0.869 ± 0.004 |
| | RL | 0.2222 ± 0.013 | 0.288 ± 0.001 | 0.285 ± 0.009 | 0.197 ± 0.003 | **0.184 ± 0.003** |
| | CV | **0.3006 ± 0.016** | 0.499 ± 0.002 | 0.492 ± 0.007 | 0.344 ± 0.005 | 0.332 ± 0.002 |

**Table 4**
The results of AP, Macro-F1, Micro-F1, and SA evaluation metrics (↑) of each comparison algorithm on 7 data sets.

| Dataset | metric | SIMM | ICML | iMvWL | VLSF | CDMM |
|---|---|---|---|---|---|---|
| Emotions | AP | 0.780 ± 0.027 | 0.578 ± 0.022 | 0.584 ± 0.015 | 0.621 ± 0.029 | **0.790 ± 0.019** |
| | Macro-F1 | 0.386 ± 0.038 | 0.182 ± 0.005 | 0.183 ± 0.004 | 0.105 ± 0.009 | **0.643 ± 0.028** |
| | Micro-F1 | 0.430 ± 0.040 | 0.404 ± 0.020 | 0.394 ± 0.014 | 0.246 ± 0.034 | **0.665 ± 0.022** |
| | SA | 0.233 ± 0.032 | 0.106 ± 0.097 | 0.009 ± 0.000 | 0.037 ± 0.017 | **0.294 ± 0.043** |
| Yeast | AP | 0.765 ± 0.016 | 0.708 ± 0.014 | 0.704 ± 0.011 | 0.631 ± 0.006 | **0.781 ± 0.009** |
| | Macro-F1 | 0.282 ± 0.006 | 0.280 ± 0.005 | 0.340 ± 0.013 | 0.324 ± 0.014 | **0.450 ± 0.015** |
| | Micro-F1 | 0.575 ± 0.010 | 0.580 ± 0.014 | 0.594 ± 0.008 | 0.442 ± 0.018 | **0.693 ± 0.011** |
| | SA | 0.105 ± 0.019 | 0.004 ± 0.002 | 0.008 ± 0.004 | 0.074 ± 0.009 | **0.207 ± 0.011** |
| Pascal07 | AP | **0.786 ± 0.005** | 0.460 ± 0.025 | 0.660 ± 0.013 | 0.767 ± 0.007 | 0.759 ± 0.006 |
| | Macro-F1 | **0.538 ± 0.009** | 0.073 ± 0.032 | 0.413 ± 0.024 | 0.523 ± 0.008 | 0.474 ± 0.018 |
| | Micro-F1 | 0.616 ± 0.009 | 0.340 ± 0.020 | 0.502 ± 0.010 | **0.623 ± 0.007** | 0.603 ± 0.014 |
| | SA | **0.394 ± 0.010** | 0.038 ± 0.010 | 0.099 ± 0.012 | 0.356 ± 0.003 | 0.359 ± 0.013 |
| Corel5k | AP | 0.534 ± 0.006 | 0.258 ± 0.004 | 0.274 ± 0.003 | 0.476 ± 0.011 | **0.545 ± 0.007** |
| | Macro-F1 | 0.098 ± 0.006 | 0.008 ± 0.001 | 0.017 ± 0.001 | 0.116 ± 0.003 | **0.173 ± 0.014** |
| | Micro-F1 | 0.320 ± 0.009 | 0.214 ± 0.007 | 0.235 ± 0.004 | 0.387 ± 0.018 | **0.485 ± 0.017** |
| | SA | 0.032 ± 0.007 | 0.000 ± 0.000 | 0.001 ± 0.001 | 0.034 ± 0.008 | **0.069 ± 0.010** |
| ESPgame | AP | 0.378 ± 0.013 | 0.219 ± 0.013 | 0.237 ± 0.002 | 0.336 ± 0.003 | **0.400 ± 0.003** |
| | Macro-F1 | 0.072 ± 0.013 | 0.013 ± 0.004 | 0.014 ± 0.000 | 0.029 ± 0.004 | **0.169 ± 0.008** |
| | Micro-F1 | 0.194 ± 0.017 | 0.206 ± 0.012 | 0.216 ± 0.002 | 0.182 ± 0.013 | **0.347 ± 0.007** |
| | SA | 0.009 ± 0.001 | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.007 ± 0.002 | **0.013 ± 0.001** |
| Iaprtc12 | AP | 0.401 ± 0.017 | 0.204 ± 0.000 | 0.242 ± 0.001 | 0.382 ± 0.002 | **0.432 ± 0.004** |
| | Macro-F1 | 0.065 ± 0.010 | 0.010 ± 0.001 | 0.021 ± 0.000 | 0.060 ± 0.002 | **0.183 ± 0.012** |
| | Micro-F1 | 0.165 ± 0.014 | 0.197 ± 0.002 | 0.223 ± 0.002 | 0.254 ± 0.006 | **0.384 ± 0.011** |
| | SA | 0.002 ± 0.001 | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.005 ± 0.001 | **0.018 ± 0.002** |
| Mirflickr | AP | **0.142 ± 0.009** | 0.093 ± 0.001 | 0.094 ± 0.005 | 0.091 ± 0.003 | 0.102 ± 0.002 |
| | Macro-F1 | 0.001 ± 0.000 | 0.007 ± 0.000 | 0.004 ± 0.001 | **0.011 ± 0.001** | 0.009 ± 0.001 |
| | Micro-F1 | 0.005 ± 0.002 | **0.082 ± 0.002** | 0.077 ± 0.004 | 0.046 ± 0.005 | 0.066 ± 0.006 |
| | SA | **0.250 ± 0.007** | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.207 ± 0.012 | 0.198 ± 0.009 |

- Compared with SIMM, CDMM has 6 metrics advantages in 8 metrics. The main reason is that CDMM pays more attention to the consistency and diversity of learning among various views when constructing a unified neural network model, and strengthens the information communication among views.
- Although the performance of CDMM is not optimal in the HL and CV metrics, combined with the SA metric, we can find that although CDMM is not as good as SIMM in the prediction results on all labels, it has achieved better results on SA. These show that CDMM has a better ability to recognize rare labels than SIMM.




(a) Hamming Loss. (b) One Error. (c) Ranking Loss.

(d) Coverage. (e) Average Precision. (f) Macro F1.

(g) Micro F1. (h) Subset Accuracy.

**Fig. 1.** Nemenyi test results for different evaluation metrics.

### 4.5. Component analysis

To verify the effectiveness of each part of CDMM, we additionally conducted component analysis experiments on the *Emotions*, *Yeast*, and *Corel5k* data sets (the parameter values involved have been given in advance), and reported the values under 8 metrics in Fig. 2. Definition of various variant algorithms corresponding to CDMM: CDMM-WD means that label correlation is not considered; CDMM-WL means ignoring view diversity information(maintain the basic structure of the model $\alpha = 0$); CDMM-DL means ignoring the influence of view contribution weight.

The results of the various sub-graphs in Fig. 2, it can be observed that CDMM is superior to its different variant algorithms on multiple metrics. Specifically, CDMM-WD and CDMM-DL achieve better results in most cases than CDMM-WL, which indicates the effectiveness of view diversity learning. CDMM achieves better results in most cases than CDMM-DL, which proves the importance of considering the difference in view contribution in the later fusion model. CDMM-WD has limited performance improvement compared with CDMM-WL and CDMM-DL, which shows that we have room for further improvement in the modeling of the correlation among labels. These results confirm the rationality and effectiveness of the modeling of each part of our model, and also clarify our motivation to use the consistency and diversity of the data from multiple views.

### 4.6. Parameter sensitivity analysis

CDMM has two essential parameters $\alpha$ and $\lambda$, which respectively control the regularization term of view-diversity and the regularization term of view-contribution. We tested the sensitivity of the parameters $\alpha$ and $\lambda$ in the range of $\{10^{-10}, 10^{-9}, \ldots, 10^{-2}\}$ and $\{10^{3}, 10^{4}, \ldots, 10^{11}\}$ respectively and reported the HL and SA results under the *Yeast* data in Fig. 3.

From Fig. 3, we can observe that when $\alpha$ is about $10^{-6}$ and $\lambda$ is about $10^{4}$, CDMM obtains relatively good performance. Besides, when $\alpha$ tends to a larger value, the performance of CDMM will decrease, mainly because an excessively large $\alpha$ value digs out more view-diversity information, but also loses some vital view consistency information. The $\lambda$ value tends to achieve the intermediate value so that each view provides enough sufficient information as much as possible. The results of other data sets and evaluation metrics are similar, and similar conclusions can be obtained.

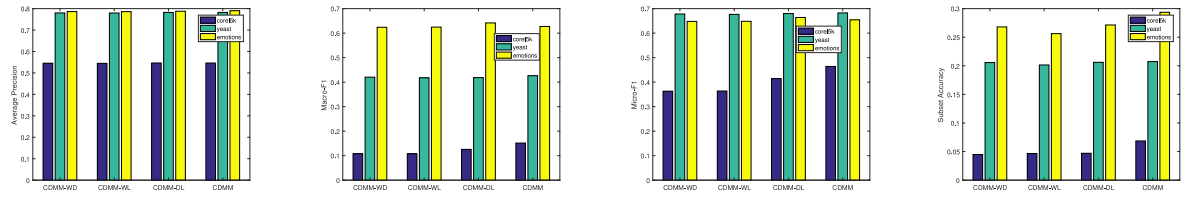
In addition, we also experimentally studied the sensitivity of $\eta$ to CDMM. Fig. 4 reports the HL and SA values of CDMM on the *Yeast* dataset, with $\eta$ values varying from 0.1 to 0.8. It can be seen that the performance of CDMM improves with the increase of $\eta$ and then decreases when $\eta > 0.6$. Without losing generality, we set the value of $\eta$ in the range of $\{0.5, 0.6, 0.7, 0.8\}$ on all datasets.

**Table 5**
Summary of the Friedman Statistics $F_F$ ($k = 5, N = 7$) and the critical value in terms of each evaluation metric ($k$: Comparing Algorithms; $N$: Data sets).

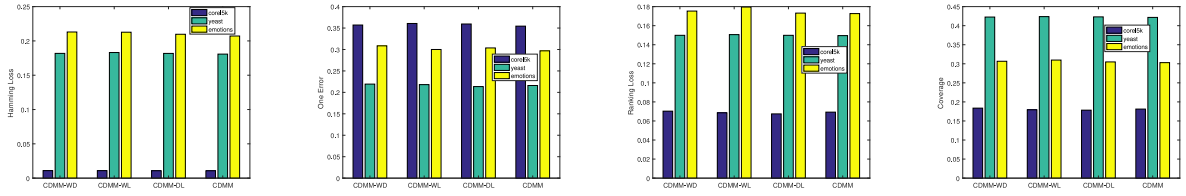
| Metric | $F_F$ | Critical Value($\rho = 0.05$) |
|---|---|---|
| Hamming Loss | 23.5477 | 2.776 |
| Average Precision | 16.6154 | |
| One Error | 17.3333 | |
| Ranking Loss | 25.4439 | |
| Coverage | 32.6842 | |
| Macro-F1 | 7.1250 | |
| Micro-F1 | 3.7351 | |
| Subset Accuracy | 19.4545 | |

## 5. Conclusion

In this paper, we have studied how to mine the information of view-consistency and view-diversity in multi-view data to achieve effective multi-view multi-label classification. For this reason, a multi-view multi-label learning framework called CDMM is proposed. It uses a unified feedforward neural network model to find out the consistency and diversity among heterogeneous views and additionally considers label correlation factors and view contribution factors. The difference from previous studies is the case that our classifier introduces a nonlinear kernel mapping function into the model. Experiments on multiple benchmark data sets have verified that the CDMM model is concise and efficient, and overall it is superior to relate to competitive solutions.




(a) Average Precision. (b) Macro-F1. (c) Micro-F1. (d) Subset Accuracy.




(e) Hamming Loss. (f) One Error. (g) Ranking Loss. (h) Coverage.

**Fig. 2.** Eight evaluation metrics results of CDMM and its variants in the *Emotions*, *Yeast* and *Core*15*k* datasets.

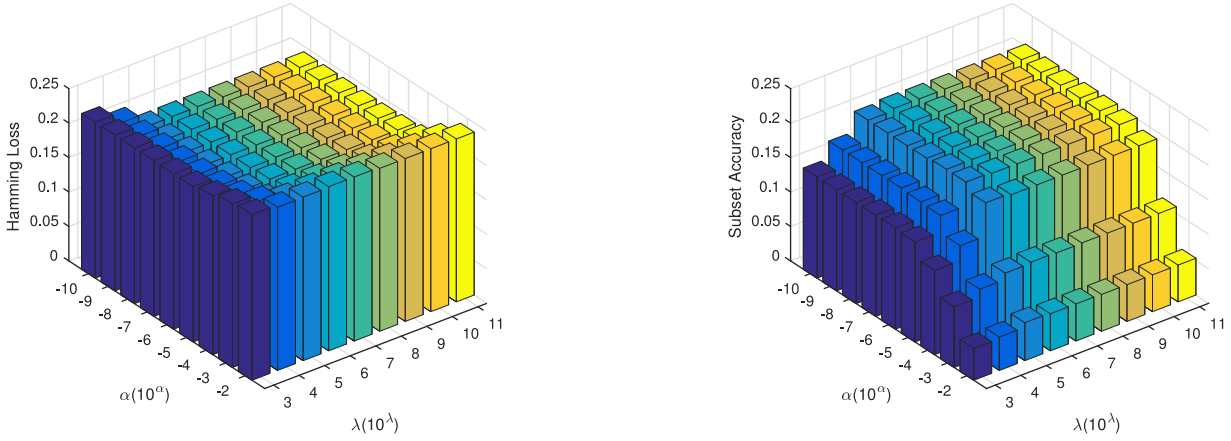



**Fig. 3.** Parameter sensitivity analysis w.r.t. $\alpha$ and $\lambda$ on *Yeast*.

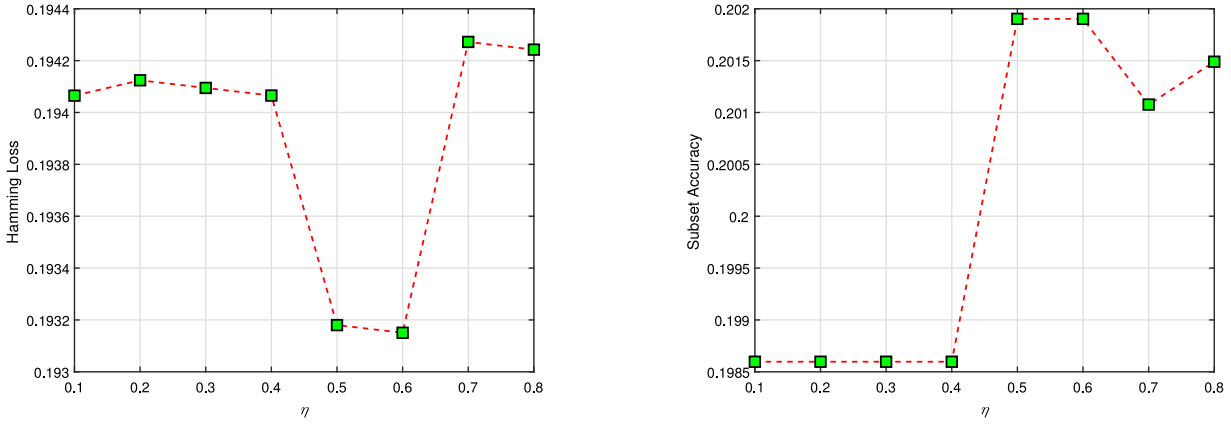



**Fig. 4.** Parameter sensitivity analysis w.r.t. $\eta$ on *Yeast*.

The disadvantage of CDMM is apparent that it is not suitable for learning in the context of incomplete data. In the future, we will expand CDMM to the application of missing feature data and label data.

## CRediT authorship contribution statement

**Dawei Zhao:** Conceptualization, Methodology, Software, Investigation, Data curation, Writing - original draft, Writing - review & editing. **Qingwei Gao:** Validation, Supervision, Project administration, Funding acquisition. **Yixiang Lu:** Visualization, Investigation. **Dong Sun:** Formal analysis, Validation. **Yusheng Cheng:** Writing - review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This research is supported by Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education (Anhui University), China (2020A003), and the Nature Science Foundation of Anhui (2008085MF183).

## References

[1] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, Christopher M. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (9) (2004) 1757–1771.
[2] S. Wen, W. Liu, Y. Yang, P. Zhou, Z. Guo, Z. Yan, Y. Chen, T. Huang, Multilabel image classification via feature/label co-projection, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2020) 1–10, http://dx.doi.org/10.1109/TSMC.2020.2967071.
[3] Feng Kang, Rong Jin, Rahul Sukthankar, Correlated label propagation with application to multi-label learning, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition vol. 2, CVPR'06, IEEE, 2006, pp. 1719–1726.
[4] André Elisseeff, Jason Weston, A kernel method for multi-labelled classification, in: Advances in Neural Information Processing Systems, 2002, pp. 681–687.
[5] Minling Zhang, Zhihua Zhou, Multilabel neural networks with applications to functional genomics and text categorization, IEEE Trans. Knowl. Data Eng. 18 (10) (2006) 1338–1351.
[6] Hao Wang, Yan Yang, Bing Liu, Hamido Fujita, A study of graph-based system for multi-view clustering, Knowl.-Based Syst. 163 (2019) 1009–1019.
[7] Zhe Xue, Guorong Li, Qingming Huang, Joint multi-view representation and image annotation via optimal predictive subspace learning, Inform. Sci. 451 (2018) 180–194.
[8] Pengfei Zhu, Qi Hu, Qinghua Hu, Changqing Zhang, Zhizhao Feng, Multi-view label embedding, Pattern Recognit. 84 (2018) 126–135.
[9] Zesen Chen, Xuan Wu, Qingguo Chen, Yao Hu, Minling Zhang, Multi-view partial multi-label learning with graph-based disambiguation, in: AAAI, 2020, pp. 3553–3560.
[10] Y. Zhang, J. Wu, Z. Cai, P.S. Yu, Multi-view multi-label learning with sparse feature selection for image annotation, IEEE Transactions on Multimedia 22 (11) (2020) 2844–2857, http://dx.doi.org/10.1109/TMM.2020.2966887.
[11] Xiaofeng Zhu, Xuelong Li, Shichao Zhang, Block-row sparse multiview multilabel learning for image classification, IEEE Trans. Cybern. 46 (2) (2015) 450–461.
[12] Meng Liu, Yong Luo, Dacheng Tao, Chao Xu, Yonggang Wen, Low-rank multi-view learning in matrix completion for multi-label image classification, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2778–2784.
[13] Xuran Zhao, Nicholas Evans, Jean-Luc Dugelay, A subspace co-training framework for multi-view clustering, Pattern Recognit. Lett. 41 (2014) 73–82.
[14] Shirui Luo, Changqing Zhang, Wei Zhang, Xiaochun Cao, Consistent and specific multi-view subspace clustering, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[15] Changqing Zhang, Ziwei Yu, Qinghua Hu, Pengfei Zhu, Xinwang Liu, Xiaobo Wang, Latent semantic aware multi-view multi-label classification, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[16] Weijieying Ren, Lei Zhang, Bo Jiang, Zhefeng Wang, Guangming Guo, Guiquan Liu, Robust mapping learning for multi-view multi-label classification with missing labels, in: International Conference on Knowledge Science, Engineering and Management, Springer, 2017, pp. 543–551.
[17] Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, Hua Zhang, Diversity-induced multi-view subspace clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 586–594.
[18] Shixing Yao, Guoxian Yu, Jun Wang, Carlotta Domeniconi, Xiangliang Zhang, Multi-view multiple clustering, 2019, arXiv preprint arXiv:1905.05053.
[19] Arthur Gretton, Olivier Bousquet, Alex Smola, Bernhard Schölkopf, Measuring statistical dependence with Hilbert-Schmidt norms, in: International Conference on Algorithmic Learning Theory, Springer, 2005, pp. 63–77.
[20] Minling Zhang, Zhihua Zhou, ML-KNN: A lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048.
[21] Minling Zhang, Lei Wu, Lift: Multi-label learning with label-specific features, IEEE Trans. Pattern Anal. Mach. Intell. 37 (1) (2014) 107–120.
[22] Suping Xu, Xibei Yang, Hualong Yu, Dong-Jun Yu, Jingyu Yang, Eric C.C. Tsang, Multi-label learning with label-specific feature reduction, Knowl.-Based Syst. 104 (2016) 52–61.
[23] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, Klaus Brinker, Multilabel classification via calibrated label ranking, Mach. Learn. 73 (2) (2008) 133–153.
[24] Yu Zhang, Dit-Yan Yeung, Multilabel relationship learning, ACM Trans. Knowl. Discov. Data (TKDD) 7 (2) (2013) 1–30.
[25] Linli Xu, Zhen Wang, Zefan Shen, Yubo Wang, Enhong Chen, Learning low-rank label correlations for multi-label classification with missing labels, in: 2014 IEEE International Conference on Data Mining, IEEE, 2014, pp. 1067–1072.
[26] Zhifen He, Ming Yang, Yang Gao, Huidong Liu, Yilong Yin, Joint multi-label classification and label correlations with missing labels and feature selection, Knowl.-Based Syst. 163 (2019) 145–158.
[27] Jun Huang, Feng Qin, Xiao Zheng, Zekai Cheng, Zhixiang Yuan, Weigang Zhang, Qingming Huang, Improving multi-label classification with missing labels by learning label-specific features, Inform. Sci. 492 (2019) 124–146.
[28] Lijuan Sun, Ping Ye, Gengyu Lyu, Songhe Feng, Guojun Dai, Hua Zhang, Weakly-supervised multi-label learning with noisy features and incomplete labels, Neurocomputing 413 (2020) 61–71.
[29] Shengjun Huang, Zhihua Zhou, Multi-label learning by exploiting label correlations locally, in: Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.
[30] Jun Huang, Xiwen Qu, Guorong Li, Feng Qin, Xiao Zheng, Qingming Huang, Multi-view multi-label learning with view-label-specific features, IEEE Access 7 (2019) 100979–100992.
[31] Qiaoyu Tan, Guoxian Yu, Carlotta Domeniconi, Jun Wang, Zili Zhang, Multi-view weak-label learning based on matrix completion, in: Proceedings of the 2018 SIAM International Conference on Data Mining, SIAM, 2018, pp. 450–458.
[32] Qiaoyu Tan, Guoxian Yu, Carlotta Domeniconi, Jun Wang, Zili Zhang, Incomplete multi-view weak-label learning, in: IJCAI, 2018, pp. 2703–2709.
[33] Changming Zhu, Panhong Wang, Lin Ma, Rigui Zhou, Lai Wei, Global and local multi-view multi-label learning with incomplete views and labels, Neural Comput. Appl. 32 (18) (2020) 15007–15028.
[34] Zhige Xie, Kai Xu, Wen Shan, Ligang Liu, Yueshan Xiong, Hui Huang, Projective feature learning for 3D shapes with multi-view depth images, in: Computer Graphics Forum, vol. 34, Wiley Online Library, 2015, pp. 1–11.
[35] Q. Tan, G. Yu, J. Wang, C. Domeniconi, X. Zhang, Individuality- and commonality-based multiview multilabel learning, IEEE Trans. Cybern. (2019) 1–12.
[36] Xuan Wu, Qingguo Chen, Yao Hu, Dengbao Wang, Xiaodong Chang, Xiaobo Wang, Minling Zhang, Multi-view multi-label learning with view-specific information extraction, in: IJCAI, 2019, pp. 3884–3890.
[37] Fangwen Zhang, Xiuyi Jia, Weiwei Li, Tensor based multi-view label enhancement for multi-label learning, in: IJCAI, 2020.
[38] Jia Zhang, Candong Li, Zhenqiang Sun, Zhiming Luo, Changen Zhou, Shaozi Li, Towards a unified multi-source-based optimization framework for multi-label learning, Appl. Soft Comput. 76 (2019) 425–435.
[39] Lu Sun, Mineichi Kudo, Keigo Kimura, Multi-label classification with meta-label-specific features, in: 2016 23rd International Conference on Pattern Recognition (ICPR), IEEE, 2016, pp. 1612–1617.
[40] Clara Pizzuti, A multi-objective genetic algorithm for community detection in networks, in: 2009 21st IEEE International Conference on Tools with Artificial Intelligence, IEEE, 2009, pp. 379–386.
[41] Jie Lu, Junyu Xuan, Guangquan Zhang, Xiangfeng Luo, Structural property-aware multilayer network embedding for latent factor analysis, Pattern Recognit. 76 (2018) 228–241.
[42] Gao Huang, Guangbin Huang, Shiji Song, Keyou You, Trends in extreme learning machines: A review, Neural Netw. 61 (2015) 32–48.
[43] Wouter F. Schmidt, Martin A. Kraaijveld, Robert P.W. Duin, et al., Feed forward neural networks with random weights, in: International Conference on Pattern Recognition, IEEE Computer Society Press, 1992, p. 1.
[44] Yoh-Han Pao, Gwang-Hoon Park, Dejan J. Sobajic, Learning and generalization characteristics of the random vector functional-link net, Neurocomputing 6 (2) (1994) 163–180.

[45] Yusheng Cheng, Dawei Zhao, Yibin Wang, Gensheng Pei, Multi-label learning with kernel extreme learning machine autoencoder, Knowl.-Based Syst. 178 (2019) 1–10.
[46] Donald W. Marquaridt, Generalized inverses, ridge regression, biased linear estimation, and nonlinear estimation, Technometrics 12 (3) (1970) 591–612.
[47] Minling Zhang, Zhihua Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng. 26 (8) (2013) 1819–1837.
[48] Janez Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.