Applied Soft Computing 124 (2022) 109071

Contents lists available at ScienceDirect

Applied Soft Computing

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

Learning view-specific labels and label-feature dependence maximization for multi-view multi-label classification

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

[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

ARTICLE INFO

Article history:
Received 22 September 2021
Received in revised form 17 May 2022
Accepted 24 May 2022
Available online 31 May 2022

Keywords:
Multi-view multi-label learning
View-specific label
Non-aligned views
Label-features dependence

ABSTRACT

Multi-view multi-label learning tasks often appear in various critical data classification scenarios. Each training sample has multiple heterogeneous data views associated with multiple labels in this learning framework simultaneously. Nevertheless, most existing methods do not consider that a single view cannot fully predict all unknown labels caused by non-aligned views, which leads to insufficient consideration of the relationship between the features and labels of each view, and the learning effect is not ideal. In this paper, we develop a novel method that uses view-specific labels and label-feature dependence maximization. Concretely, we first assume that each view and its corresponding label space have a smooth local structure. In this way, the view-specific label learning model is constructed, enhancing the consistency and complementarity of label space information. Then, multiple multi-label classifiers are constructed by maximizing label-feature dependence. Finally, the linear classification model is extended to the nonlinear, and the prediction stage is combined with the contribution weight of each view. The results of several benchmark datasets show that our proposed method is significantly more effective than the state-of-the-art methods.

1. Introduction

With the rapid development of data collection technology in big data, objects often have features from different sources, that is, the feature representation of multiple views [1–3]. For example, video data can be represented by three various forms of data at the same time: text, image, and sound (the diversity of data collection); for pictures, we can use different characteristics to describe (the variety of data description), such as texture description, shape description, color description, surrounding text.

Multi-view(MV) learning methods have been developed and paid more attention, but previous studies have mainly focused on where each target object has only a single label [4–6]. Each object may contain multiple semantic information in the real world, such as multiple labels that simultaneously annotate a picture with semantic information: blue sky, sea, and fish. It can be observed that there are interdependencies among labels in multi-label(ML) learning. However, a basic assumption of the traditional single-label multi-view learning method is that the labels are mutually exclusive, which conflicts with the multi-label learning problem. Therefore, traditional multi-view learning can no longer solve the multi-semantic problem of multi-source heterogeneous data. Researchers have proposed that the multi-view multi-label framework has been applied in many fields, such as image annotation [7,8], medical diagnosis [9], and video analysis [10].

A simple solution to the multi-view multi-label(MVML) problem is to consider concatenating the data of multiple views to degenerate the MVML problem into a multi-label problem. But it ignores the inherent physical meaning of each view, and high-dimensional data may cause unique catastrophic problems and over-fitting problems. Another method is to perform parallel learning for each view and then directly merge multiple views when predicting. However, it ignores the supplementary information among the views and does not consider the impact of the different contributions of each view. Consistency can explicitly use the corresponding information hidden in the original multi-view features, and complementarity can describe the difference among various views [11]. Therefore, how to use the consistency and complementary information between different views in MVML learning is still a challenging problem, which can obtain better performance.

In multi-view learning, it is usually assumed that multiple views are aligned, but in reality, it is usually challenging to associate multiple views with each other due to data privacy protection considerations. For example, in the recommendation system, user information among multiple systems cannot be effectively connected and interacted for data security and privacy

* 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.asoc.2022.109071
1568-4946/

protection [12]. The previous method did not explicitly specify the unique label information hidden in each view. Due to the widespread existence of such non-aligned views [13], each view can only obtain part of the information in the complete label space in MVML learning, and a single view cannot characterize all relevant labels fully.

A common assumption is that similar features in instance space usually share the same subset of labels. Thus, manifold regularization can explore the local structural properties between different view features and corresponding labels. For example, a picture marked with a blue sky label should have a blue sky feature, and vice versa, an image with a blue sky feature should also have a blue sky label. However, the label information of each view is inconsistent, and the multi-view framework of the unified label set cannot effectively reflect the problem of the relevance of label features. It can observe that the existing multi-view learning methods do not fully integrate the dependent knowledge between labels and features.

Based on the above research, we propose a **M**ulti-view multi-label classification method via learning **V**iew-specific **L**abels and label-feature **D**ependence maximization(MVLD). MVLD takes a two-stage strategy: In the first stage, a novel view-specific label learning method is proposed to solve the label inconsistency problem caused by non-aligned view information. In the second stage, we first maximize the label-feature dependence based on the label smoothness assumption, which solves the problem of view consistency, and then assign weights to each view to solve the problem of view complementarity. The multi-label prediction model is constructed through the above two-stage strategy. The main contributions of this paper are summarized as follows:

1. As far as we know, we propose a novel MVML learning method, which is the first attempt to learn view-specific labels in the MVML framework, which is utilized to solve the problem of non-aligned views.
2. We use the topological structure information of the view feature to learn the view-specific labels to solve complementary view information. The Hilbert–Schmidt Independence Criterion(HSIC) theory is utilized to maximize the dependence between the view-specific labels and the characteristics of each view. The unified multi-label learning model is constructed to solve the consistency problem of the view information.
3. We extend the linear model to the nonlinear model to solve the situation where the given data cannot be linearly separable.

2. Related work

Our work involves two aspects: multi-view learning and multi-label learning. This section will briefly review some state-of-the-art methods in these two areas.

2.1. Multi-label learning

In many computer vision [14,15], natural language learning [16], and text classification [17] tasks, there is an issue that an example is associated with multiple labels simultaneously. Multi-label learning is a learning framework that deals with multi-semantic problems in objects, which aims to learn a model that can assign appropriate label sets to unseen examples through training data. According to different task types, multi-label learning problems are divided into problem transformation(PT) and algorithm adaptation(AA) methods.

PT method: Transform the multi-label learning problem into other known learning problems, such as two or multi-type, and label ranking problems. BR(binary relevance) [15] decomposes the multi-label problem into independent binary classification tasks and uses a single-label learning method to solve the multi-label learning problem. RAkEL(random k-label sets) [18] transforms the multi-label learning problem into an integrated multi-class learning problem. CLR(calibrated label ranking) [19] transforms the multi-label learning problem into a calibration label ranking problem through the pairwise comparison of labels.

AA Method: Directly design a multi-label method to solve the problem of instances associated with multiple labels. The RankSVM(A kernel method for multi-labeled classification) [20] extends the classic support vector machine (SVM) to the multi-label learning problem, which can minimize the ranking loss and perform quadratic programming on each category label's SVM classification result to obtain the multi-label classification result. MLkNN(multi-label learning for k-nearest neighbor) [21] uses the Maximum A Posteriori(MAP) estimation criterion to predict its category label set according to the label information of the k nearest neighbor samples of each sample. Based on the naive Bayesian multi-label classification algorithm [22], the feature selection mechanism is added to the algorithm. Zhang et al. [23] uses K-means to cluster the labels in positive and negative directions to understand the specific characteristics of each label and then combines the SVM classifiers of each label through a binary classifier. Furthermore, Huang et al. [24] obtains label-specific features(LLSF) by constructing a least-squares loss term and integrating the correlation information between labels and the sparsity of weight coefficients.

As mentioned above, multi-label learning methods can be directly applied to MVML learning via feature sequences or parallel connections but rarely consider the consistency and complementarity between different views.

2.2. Multi-view multi-label learning

Most of the previous multi-label learning algorithms assume that the data come from the same view data, but there are many methods and routes to obtain data in the real world. In real application scenarios, a single view cannot accurately describe the semantic information of the target object. To solve the above problems, some literatures propose optimization solutions for MVML, such as:

1. Focusing on the multi-label learning method: Huang et al. [25] learns view consistency and complementary information through label association and view contribution, and view label-specific feature learning(VLSF) is utilized to build MVML classifiers. Similarly, Chen et al. [26] first uses the fusion similarity map to eliminate the ambiguity of the candidate labels, then guides the specific features of the labels generated by the cluster analysis, and finally combines the SVM classifier to learn the prediction model of each label. Ren et al. [27] fuses multiple views into a mixed feature matrix and uses a low-rank structure and manifold regularization to utilize global label correlation and local smoothness further.
2. Focusing on the multi-view learning method: Liu et al. [28] first learns the low-dimensional common representation of all views through the subspace learning(lrMMC), then restricts it to low rank to adapt to matrix completion, and finally learns the contribution weight of different views to explore the complementarity between different views. Zhang et al. [29] first retains the high-order relationship between different viewpoints through tensor factorization and then mines a more comprehensive public representation topology and transfers it to the label space. Tan et al. [30] learns a distinguished shared subspace

from incomplete views(iMvML) through nonnegative matrix factorization and learns a robust weak label classifier combined with local label structure. Zhang et al. [31] learns latent semantic information through matrix factorization and aligns it with benchmarks to obtain consensus representation information that encodes complementarity and consistency. Tan et al. [32] learns the individuality information and common information about different views(ICM2L) through nonnegative matrix factorization to learn the complementary knowledge and consistent communication. Fang et al. [33] through iterative learning of multi-view features mapping matrix and consistent feature view representation.

The above methods can solve the problem of MVML to a certain extent but are primarily based on linear models. Thus, the experiment may not achieve the expected results when the data is linearly inseparable. In order to handle this problem, scholars have added nonlinear mapping methods to the model. For example, Zhao et al. [34] learns a common representation matrix among views based on subspaces, uses HSIC theory to constrain the inconsistency among private matrices, builds a two-step stage multi-label classifier(TM3L) based on label correlation, and finally extends the model to nonlinear models. Zhu et al. [35] first connects the label space and feature space of different views by a latent space, then uses the HSIC to explore the consistency among different views, and finally expands the model. Wu et al. [36] adds nonlinear mapping to the neural network structure, which utilizes the development of shared subspaces and view-specific information to learn view consistency and complementary information. On this basis, Shen et al. [9] introduces label correlations information and encodes view-specific information into latent semantics to strike a balance with label correlations. Zhao et al. [11] builds multiple multi-label models through the kernel method to learn view-consistency and view-diversity information(CDMM).

As mentioned above, most MVML methods assume that all views have a unified label set, but each view can only obtain part of the overall information in practical applications. A single view cannot fully predict all the unknown labels, so different views have their specific labels collection. There is currently no algorithm for learning view-specific labels based on our knowledge. Although learning common representations through subspace methods can avoid this problem, it becomes more challenging to learn effective latent low-dimensional consensus representations as to the number of views increases. Besides, each view of data has heterogeneity, and a simple linear model can no longer meet the needs of MVML learning. To this end, we propose a view-specific label and label-feature dependence maximization MVML learning framework. Fig. 1 illustrates the model framework of the proposed algorithm. First, we learn the specific label of each view through the topology of each view. Then view-specific labels are introduced into the multi-label learning framework, and the interdependence between each view feature and labels is enhanced through HSIC, and we combine the contributions of each view to learn the complementary information between the views. Finally, MVLD is extended to the nonlinear model. Compared with the state-of-the-art methods, MVLD has excellent competitiveness on various benchmarks data sets.

3. Proposed approach

3.1. Problem statement and notations

Let $\boldsymbol{X} = \{\boldsymbol{x}_v\}_{v=1}^{m}$ denote an MVML data sets with m views, where $\boldsymbol{X}_v = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_N]^{\mathrm{T}} \in \mathbb{R}^{N\times d_v}$ is the complete feature space of the vth view. $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_N] \in \mathbb{R}^{N\times q}$ represents the label space corresponding to the feature set, where $\boldsymbol{y}_i \in \{0, 1\}^{N\times q}$ is the label vector of $\boldsymbol{x}_i$, N is the number of samples, and q represents the number of labels. For clarity, we define the symbols used in this paper and their corresponding descriptions in Table 1.

Table 1
The symbols and descriptions.

Notations	Description
$\boldsymbol{X}_v$	The vth view feature set
$\boldsymbol{Y}$	Multi-view shared label set
$\boldsymbol{U}_v$	The vth view specific label matrix
$\boldsymbol{A}_v$	The vth view weight coefficient matrix
$\boldsymbol{L}_v$	The vth view graph Laplacian matrix of instance correlation matrix $\boldsymbol{S}$
$\boldsymbol{\theta}_v$	The vth view prediction contribution weight matrix
$\boldsymbol{K}_v$	The vth view kernel matrix
$\boldsymbol{N}$	The number of training samples
$\boldsymbol{d}_v$	The vth view number of features
$\boldsymbol{q}$	The number of labels

3.2. Multi-view multi-label basic model construction

We construct a basic linear classification model through a least-squares loss for the learning task of the MVML classification fusion strategy. The underlying classification model associates different views with the same set of labels to predict unknown instance labels, which can use consistent information across different views. The specific model is defined as follows:

$$\min_{\boldsymbol{W}_v} \frac{1}{2}\sum_{v=1}^{m} \|\boldsymbol{X}_v\boldsymbol{W}_v - \boldsymbol{Y}\|_F^2 + \lambda R\left(\boldsymbol{W}_v\right) \tag{1}$$

where $\boldsymbol{W}_v$ is the weight matrix of the linear model of the vth view, $R\left(\boldsymbol{W}_v\right)$ is a regularization term, λ is the regularization parameter, and $\|\cdot\|_F$ represents the *Frobenius* norm. Eq. (1) is not suitable for multi-view multi-label data, and the main reasons are:

1. Although this model can solve the influence of unaligned views to a certain extent, it ignores the inconsistency of labels corresponding to views in unaligned views.
2. The model ignores the dependencies between features and labels. There are often certain local structural characteristics between the features and labels of instances in MVML data. In short, if the features between two instances are similar, then their corresponding labels are also similar.

The main issue that needs attention in the next section is how to effectively solve these two limitations and make our model more robust and discriminative.

3.3. View-specific labels information extraction

The traditional MVML learning task assigns a unified label set to all views. An effective strategy in MVML learning is to use ensemble learning, but the separate mode of ensemble learning has higher requirements for the classification performance of each classifier. A unified label set will affect the classification effect when used as the prediction target of the ensemble method. In the real world, due to non-aligned views, each view can only get a part of the overall label set, and it is impractical to construct the label set for each view separately. For this reason, we consider assigning a unique label set for each view in the integrated classification learning, which will effectively improve the performance of the algorithm. Assigning a specific label to each view is conducive to mining the deep relationship between the label and the feature in each view, which can improve the performance of the classifier. However, the view-specific label

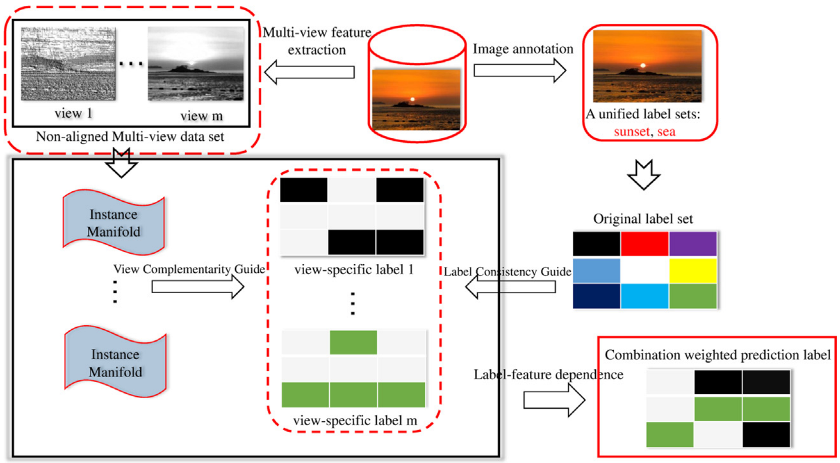


Fig. 1. The learning framework of the proposed method MVLD. Illustration of extracting multiple views feature information in instance space and a globally unified label set corresponding to the instance.

is not easy to obtain directly through prior knowledge. For this reason, we adopt a learning method to obtain the specific label of the view. Thus, the problem can be written in the following form:

$$\min_{\boldsymbol{U}_v} \sum_{v=1}^{m} \left(\|\boldsymbol{U}_v - \boldsymbol{Y}\|_F^2 \right) \tag{2}$$

where $\boldsymbol{U}_v$ represents the label set corresponding to the vth view.

For MVML data, an intuitive assumption is that the feature of each view has its unique geometric topology. Because of the close association between features and labels, there are geometric similarities between them, which provides a theoretical basis for us to learn view-specific labels from the geometric topology of features. Specifically, if two instances from the same feature space are similar, they should also have similar labels in the label space $\boldsymbol{Y}$ [8]. Based on this, we can induce learning view-specific labels through the topology information of different views. Besides, the feature representations of different views will provide complementary information, and we use the different contribution weights of each view to learn it. The local smoothing regularization term can be defined as follow:

$$\begin{aligned}\sum_{v=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{s}_v{}^{i,j}\left(\boldsymbol{u}_v^i - \boldsymbol{u}_v^j\right)^2 &= \sum_{v=1}^{m} Tr\left(\boldsymbol{U}_v{}^{\mathrm{T}}\boldsymbol{\omega}_v\left(\boldsymbol{D}_v - \boldsymbol{S}_v\right)\boldsymbol{U}_v\right) \\ &= \sum_{v=1}^{m} Tr\left(\boldsymbol{U}_v{}^{\mathrm{T}}\boldsymbol{\omega}_v\boldsymbol{L}_v\boldsymbol{U}_v\right)\end{aligned} \tag{3}$$

where $\boldsymbol{L}_v = \boldsymbol{D}_v - \boldsymbol{S}_v$ is the graph Laplacian matrix, $\boldsymbol{D}_v$ is the diagonal matrix of $\boldsymbol{d}_v^{ii} = \sum_{j=1}^{n} \boldsymbol{s}_v^{ij}$, $\boldsymbol{S}_v$ is the weight matrix, and $Tr(\cdot)$ is the trace of a matrix. The non-negative trade-off parameter $\boldsymbol{\omega}$ is a weight value between $[0-1]$, utilized to measure the contribution of each feature view when using its data local structure relationship. The higher the weight value, the greater the contribution of this heterogeneous data. $\boldsymbol{S}^{ij}$ measures the similarity between instances $\boldsymbol{X}^i$ and $\boldsymbol{X}^j$. Besides, the similarity between the two instances of the vth view is calculated by heat kernel:

$$\boldsymbol{S}_v^{i,j} = \begin{cases} e^{-\frac{\left\|x_v^i - x_v^j\right\|^2}{2\sigma^2}}, & if\ \boldsymbol{x}_v^j \in N_p\left(\boldsymbol{x}_v^i\right)\ \ or\ \ \boldsymbol{x}_v^i \in N_p\left(\boldsymbol{x}_v^j\right) \\ 0, & otherwise \end{cases} \tag{4}$$

where $N_p(\boldsymbol{x})$ is the set of p nearest neighbors of instance $\boldsymbol{x}$ obtained by Euclidean distance neighborhood search. By combining Eqs. (2) and (3), our final view-specific labels optimization problem can be expressed as:

$$\min_{\boldsymbol{\omega}_v, \boldsymbol{U}_v} \sum_{v=1}^{m} \left(\|\boldsymbol{U}_v - \boldsymbol{Y}\|_F^2 + \lambda_1 Tr\left(\boldsymbol{U}_v^{\mathrm{T}}\boldsymbol{\omega}_v\boldsymbol{L}_v\boldsymbol{U}_v\right)\right) \tag{5}$$

3.4. Label-feature dependence maximization

According to the smoothness assumption, the labels and features of the instances have local structural similarities. For example, instances labeled "sea" and "sand" should have the characteristics of "sea" and "sand" and vice versa. To adequately account for label-function dependencies, we use HSIC [37] to quantitatively describe the dependencies between labels and features.

$$(N-1)^{-2} Tr\left(\boldsymbol{HPHM}\right) \tag{6}$$

where $\boldsymbol{H}, \boldsymbol{P}, \boldsymbol{M} \in \mathbb{R}^{N\times N}$, $\boldsymbol{H}_{ij} = \delta_{ij} - 1/N$ if $i = j$ then $\delta_{ij} = 1$ otherwise $\delta_{ij} = 0$, $\boldsymbol{H}, \boldsymbol{M}, \boldsymbol{P} \in \mathbb{R}^{N\times N}$ is the feature map of the sample, and $\boldsymbol{P} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is the feature map of the label. According to the hypothesis of the similarity of the feature and the local structure of the label, when the information between the two sample features is very similar, the corresponding predicted label vector is also very similar. For further analysis, we can regard $\boldsymbol{M}$ as the semantic information of the sample, and semantic information and feature information are usually positively correlated [38]. Based on this, we maximize Eq. (6) to improve MVML learning performance from the perspective of instance semantics. Intuitively, the HSIC theory induces theoretical agreement between the prediction model and the view-specific feature model on the smoothness assumption.

We construct the view-specific labels and label-feature dependence maximization multi-label prediction model, which goal is to use the topological structure between different views to induce the learning of specific features of the view. Then our objective function can be expressed in the following form:

$$\begin{aligned}\min_{\boldsymbol{W}_v} \sum_{v=1}^{m} \Big\{ &\|\boldsymbol{X}_v\boldsymbol{W}_v - \boldsymbol{U}_v\|_F^2 - \lambda_2 Tr\left(\boldsymbol{W}_v{}^{\mathrm{T}}\boldsymbol{X}_v{}^{\mathrm{T}}\boldsymbol{G}_v\boldsymbol{X}_v\boldsymbol{W}_v\right) \\ &+ \lambda_3 \|\boldsymbol{W}_v\|_F^2 \Big\}\end{aligned} \tag{7}$$

where $\boldsymbol{G}_v = (N-1)^{-2}\boldsymbol{HP}_v\boldsymbol{H}$.

Eq. (7) is conducive to mining more sufficient information between the label features to predict the unknown label set of MVML. Furthermore, the learning model used in Eq. (7) is a linear input model, which cannot handle the nonlinear case where MVML data cannot be linearly separable. It is worth noting that many existing MVML methods (such as [25,30,32]) use linear models, and we expect to use nonlinear models to obtain better performance. In the MVML data, there is heterogeneity among the views, and the linear model cannot effectively solve this problem. This paper utilizes a kernel expansion [39,40] method to address the issues mentioned above. Concretely, we use feature mapping $\phi(\cdot)$ to map the feature space $\boldsymbol{X}$ to a higher-dimensional Hilbert space $\mathcal{H}(\cdot)$. Based on the indicator theorem [41], $\boldsymbol{W}$ can represent $\boldsymbol{W} = \phi(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{A}$ by a linear combination of input variables. $\boldsymbol{K}_{ij} = \kappa\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \phi(\boldsymbol{x})\,\phi(\boldsymbol{x})^{\mathrm{T}}$ is the kernel matrix, where $\kappa(\cdot,\cdot)$ is the kernel function(the RBF kernel function is used in this article). Thus, Eq. (7) can be rewritten as:

$$\min_{\boldsymbol{A}_v} \sum_{v=1}^{m} \left\{ \|\boldsymbol{K}_v\boldsymbol{A}_v - \boldsymbol{U}_v\|_F^2 - \lambda_2 Tr\left(\boldsymbol{A}_v{}^{\mathrm{T}}\boldsymbol{K}_v{}^{\mathrm{T}}\boldsymbol{G}_v\boldsymbol{K}_v\boldsymbol{A}_v\right) + \lambda_3 Tr\left(\boldsymbol{A}_v^{\mathrm{T}}\boldsymbol{K}_v\boldsymbol{A}_v\right)\right\} \tag{8}$$

When predicting the unknown instance label set, we add weights to consider the contribution of each view to the label prediction according to the literature [11,42]. After the optimization process is completed, the update rule for the contribution weight of each view is as follows:

$$\begin{aligned} \boldsymbol{\theta}_v &= \frac{\lambda_4 + \sum_{v=1}^{m}\boldsymbol{Q}_v - m\boldsymbol{Q}_v}{m\boldsymbol{\lambda}_4} \\ \boldsymbol{Q}_v &= \left\|\boldsymbol{K}_v^{(train)}\boldsymbol{A}_v - \boldsymbol{U}_v\right\|_F^2 \end{aligned} \tag{9}$$

Given the test sample $\widehat{\boldsymbol{X}}$, we calculate the corresponding kernel function mapping $\kappa\left(\widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{x}}_j\right)$, combined with the prediction vector corresponding to each view, and our final prediction function is expressed as:

$$\boldsymbol{Y}_{pre} = \sum_{v=1}^{m} \boldsymbol{\theta}_v \boldsymbol{K}_v^{(test)}\boldsymbol{A}_v \tag{10}$$

Then the final predicted label vector is expressed as: $\hat{\boldsymbol{Y}} = sign\left(\boldsymbol{Y}_{pre}\right)$.

MVLD not only considers the learning of view-specific labels but also considers the measurement of the dependency relationship between it and each view feature. The two are closely related before and after to improve the performance of the approach more effectively. Besides, in the prediction stage, the combined results of each view are obtained like ensemble learning, which makes the algorithm performance more robust. The following experiments will confirm these advantages.

3.5. Optimization

MVLD involves a two-stage optimization. This section will give a detailed introduction to the two-stage optimization process.

The first stage: Problem (5), involves the optimization of two variables, $\boldsymbol{U}_v$ and $\boldsymbol{\omega}_v$, and it is tough to optimize these two variables simultaneously. We use alternating optimization techniques to optimize the objective function, thereby updating each variable.

When $\boldsymbol{\omega}_v$ is fixed, the gradient of $\boldsymbol{U}_v$ can be obtained as:

$$\nabla\boldsymbol{U}_v = \left(\boldsymbol{I} + \lambda_1\boldsymbol{\omega}_v\boldsymbol{L}_v\right)\boldsymbol{U}_v - \boldsymbol{Y} \tag{11}$$

Then the closed form solution is:

$$\boldsymbol{U}_v = \left(\boldsymbol{I} + \lambda_1\boldsymbol{\omega}_v\boldsymbol{L}_v\right)^{-1}\boldsymbol{Y} \tag{12}$$

When $\boldsymbol{U}_v$ is fixed, update $\boldsymbol{\omega}_v$ as follows:

$$\boldsymbol{\omega}_v = \frac{\frac{1}{Tr\left(\boldsymbol{U}_v{}^{\mathrm{T}}\boldsymbol{L}_v\boldsymbol{U}_v\right)}}{\sum_{v=1}^{m}\frac{1}{Tr\left(\boldsymbol{U}_v{}^{\mathrm{T}}\boldsymbol{L}_v\boldsymbol{U}_v\right)}} \tag{13}$$

The second stage: the optimization of Eq. (8). We can calculate the gradient w.r.t $\boldsymbol{A}_v$ as follows:

$$\nabla\boldsymbol{A}_v = \boldsymbol{K}_v\boldsymbol{K}_v{}^{\mathrm{T}}\boldsymbol{A}_v - \boldsymbol{K}_v\boldsymbol{V}_v + \lambda_2\boldsymbol{K}_v\boldsymbol{A}_v - \lambda_3\boldsymbol{K}_v^{\mathrm{T}}\boldsymbol{G}_v\boldsymbol{K}_v\boldsymbol{A}_v \tag{14}$$

When we set the derivative w.r.t $\boldsymbol{A}_v$ to zero, we have the following formula:

$$\boldsymbol{A}_v = \left(\boldsymbol{K}_v{}^{\mathrm{T}} + \lambda_2\boldsymbol{I} - \lambda_3\boldsymbol{G}_v\boldsymbol{K}_v\right)^{-1}\boldsymbol{U}_v \tag{15}$$

Given the above process of optimizing and predicting the model, we summarize the main steps of MVLD in Algorithm 1.

Algorithm 1: **M**ulti-view multi-label classification method via learning **V**iew-specific label and **L**abel-feature **D**ependence maximization

Input: Training data matrix: $\{\boldsymbol{X}_v\}_{v=1}^{m}$, $\boldsymbol{Y} \in \{-1, 1\}^{N\times q}$;
Trade-off parameters: λ_1, λ_2, λ_3, and λ_4;
Kernel function parameter: σ;
Number of iterations: t;
Output: Predicted label matrix: $\boldsymbol{Y}_{pre}$;

1 Randomly initialize $\boldsymbol{U}_v$, $\boldsymbol{\omega}_v$ and $\boldsymbol{\theta}_v$;
2 The first stage:
3 **for** $j = 1, 2, \cdots, t$ **do**
4   **for** $v = 1, 2, \cdots, m$ **do**
5   Construct the affinity graph for the v-th view by Eq. (4) and calculate $\boldsymbol{L}_v = \boldsymbol{D}_v - \boldsymbol{S}_v$;
6   Update $\boldsymbol{U}_v$ by Eq. (12);
7   Update $\boldsymbol{\omega}_v$ by Eq. (13);
8   **if** *convergence* **then**
9   break;
10 The second stage:
11 **for** $v = 1, 2, \cdots, m$ **do**
12   Update $\boldsymbol{A}_v$ by Eq. (15);
13   Update $\boldsymbol{\theta}_v$ by Eq. (9);
14   Calculate the predicted label vector $\boldsymbol{Y}_{pre}$ by Eq. (10);

3.6. Complexity analysis

The time complexity of MVLD is mainly controlled by step 3, step 8, step 11, step 12, and step 14.

In the first stage, the time complexity of learning $\boldsymbol{U}_v$, $\boldsymbol{\omega}_v$, and constructing $\boldsymbol{L}_v$ for each iteration is $\mathcal{O}\left(N^3 + N^2q\right)$, $\mathcal{O}\left(N^2q + Nq^2\right)$, and $\mathcal{O}\left(N^2d\right)$. The overall complexity of the first stage is $\mathcal{O}\left(t\left(N^3 + N^2q + N^2d + Nq^2\right)\right)$, where t is the total number of iterations. Besides, the convergence speed of our model is quite fast, so t can be very small.

In the second stage, the overall time complexity of calculating $\boldsymbol{A}_v$ and $\boldsymbol{\theta}_v$ is $\mathcal{O}\left(N^3 + N^2q\right)$. Since $N \gg m$ normally holds, the overall time complexity of MVLD can be further reduced to $\mathcal{O}\left(t\left(N^3 + N^2d\right) + N^3\right)$.

4. Experimental content

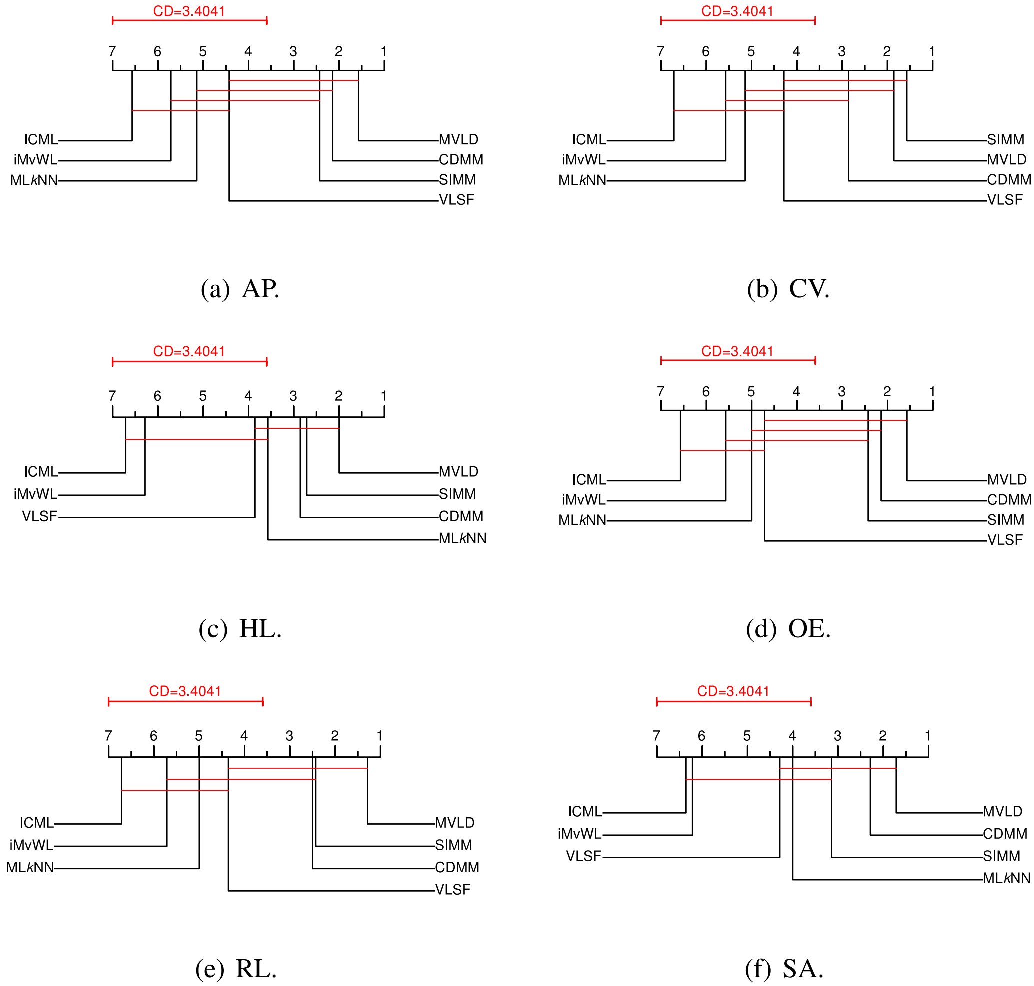
In this section, we conduct a large number of experimental studies to evaluate the effectiveness of MVLD. We compare 5 state-of-the-art approaches on 7 real-world MVML benchmark datasets based on 6 widely used evaluation metrics. Furthermore, the classical multi-label learning method MLkNN is selected and

Table 2
MVML data sets.

Views	Emotions	Yeast	Pascal07	Corel5k	Espgame	Iaprtc12	mirflickr
1	Rhythmic attributes (8)	Genetic expression (79)	DenseSift (1000)	DenseHue (100)	DenseHue (100)	DenseHue (100)	DenseHue (100)
2	Timbre attributes (64)	Phylogenetic profile (24)	HarrisSift (1000)	DenseSift (1000)	DenseSift (1000)	DenseSift (1000)	DenseSift (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
The number of labels	6	14	20	260	268	291	457
The number of samples	593	2417	9963	4999	20 770	19 627	25 000

Table 3
The experimental results (mean ± std) of each comparison approaches on 6 evaluation metrics. ↓ (↑) means that the smaller (larger) the value, the better the performance.

Dataset	AP(↑)						
	ML*k*NN	SIMM	ICM2L	iMvWL	VLSF	CDMM	MVLD
Emotions	.718 ± .021	.780 ± .027	.578 ± .022	.584 ± .015	.621 ± .029	.790 ± .019	**.803 ± .020**
Yeast	.762 ± .009	.765 ± .016	.708 ± .014	.704 ± .011	.631 ± .006	.781 ± .009	**.785 ± .006**
Pascal07	.464 ± .007	**.781 ± .004**	.460 ± .025	.660 ± .013	.767 ± .007	.759 ± .006	.767 ± .004
Corel5k	.349 ± .008	.534 ± .006	.258 ± .004	.274 ± .003	.476 ± .011	.545 ± .007	**.548 ± .003**
ESPgame	.259 ± .003	.378 ± .013	.219 ± .013	.237 ± .002	.336 ± .003	**.400 ± .003**	.396 ± .005
Iaprtc12	.340 ± .004	.401 ± .017	.204 ± .000	.242 ± .001	.382 ± .002	**.432 ± .004**	.429 ± .005
Mirflickr	.076 ± .001	**.142 ± .009**	.093 ± .001	.094 ± .005	.091 ± .003	.102 ± .002	.103 ± .002
Dataset	**CV(↓)**						
	ML*k*NN	SIMM	ICM2L	iMvWL	VLSF	CDMM	MVLD
Emotions	.376 ± .020	.307 ± .014	.530 ± .036	.506 ± .014	.441 ± .022	.304 ± .018	**.296 ± .008**
Yeast	.452 ± .006	.450 ± .004	.503 ± .006	.494 ± .009	.601 ± .008	.426 ± .008	**.424 ± .014**
Pascal07	.319 ± .003	**.109 ± .003**	.308 ± .048	.189 ± .015	.110 ± .001	.111 ± .003	.110 ± .003
Corel5k	.290 ± .004	**.148 ± .006**	.334 ± .000	.286 ± .004	.187 ± .009	.179 ± .011	.177 ± .006
ESPgame	.437 ± .003	**.308 ± .012**	.479 ± .001	.447 ± .004	.365 ± .005	.337 ± .005	.320 ± .000
Iaprtc12	.376 ± .005	**.270 ± .019**	.497 ± .001	.435 ± .003	.327 ± .002	.284 ± .006	.275 ± .002
Mirflickr	.386 ± .004	**.306 ± .016**	.499 ± .002	.492 ± .007	.344 ± .005	.332 ± .002	.326 ± .006
Dataset	**HL(↓)**						
	ML*k*NN	SIMM	ICM2L	iMvWL	VLSF	CDMM	MVLD
Emotions	.262 ± .010	.246 ± .008	.375 ± .015	.395 ± .011	.293 ± .007	.207 ± .014	**.205 ± .013**
Yeast	.196 ± .005	.207 ± .005	.278 ± .008	.269 ± .005	.258 ± .004	.189 ± .006	**.184 ± .006**
Pascal07	.072 ± .001	**.046 ± .001**	.115 ± .003	.086 ± .002	.050 ± .000	.049 ± .001	**.046 ± .001**
Corel5k	.013 ± .000	**.011 ± .000**	.022 ± .000	.022 ± .000	.013 ± .000	**.011 ± .000**	**.011 ± .000**
ESPgame	**.017 ± .000**	**.017 ± .000**	.029 ± .000	.028 ± .000	**.017 ± .000**	.018 ± .000	**.017 ± .000**
Iaprtc12	**.019 ± .000**	**.019 ± .000**	.032 ± .000	.031 ± .000	**.019 ± .000**	**.019 ± .000**	**.019 ± .000**
Mirflickr	**.006 ± .000**	**.006 ± .000**	.013 ± .000	.013 ± .000	**.006 ± .000**	**.006 ± .000**	**.006 ± .000**
Dataset	**OE(↓)**						
	ML*k*NN	SIMM	ICM2L	iMvWL	VLSF	CDMM	MVLD
Emotions	.359 ± .041	.310 ± .056	.530 ± .030	.521 ± .021	.539 ± .073	.304 ± .034	**.272 ± .050**
Yeast	.233 ± .017	.225 ± .028	.235 ± .024	.292 ± .020	.343 ± .015	.211 ± .013	**.210 ± .015**
Pascal07	.585 ± .012	**.261 ± .010**	.589 ± .002	.397 ± .021	.290 ± .014	.308 ± .011	.285 ± .008
Corel5k	.602 ± .017	.363 ± .011	.697 ± .007	.687 ± .003	.438 ± .017	.362 ± .006	**.357 ± .012**
ESPgame	.650 ± .008	.476 ± .022	.713 ± .030	.674 ± .000	.533 ± .011	**.465 ± .009**	.474 ± .010
Iaprtc12	.535 ± .006	.447 ± .021	.720 ± .005	.624 ± .002	.465 ± .006	**.439 ± .007**	.446 ± .008
Mirflickr	.905 ± .003	**.865 ± .006**	.908 ± .004	.887 ± .003	.891 ± .003	.869 ± .004	.866 ± .005
Dataset	**RL(↓)**						
	ML*k*NN	SIMM	ICM2L	iMvWL	VLSF	CDMM	MVLD
Emotions	.255 ± .020	.178 ± .024	.443 ± .013	.414 ± .012	.330 ± .022	.174 ± .013	**.163 ± .015**
Yeast	.170 ± .006	.165 ± .008	.215 ± .011	.214 ± .008	.320 ± .006	.151 ± .008	**.150 ± .006**
Pascal07	.256 ± .004	**.068 ± .003**	.241 ± .040	.138 ± .011	.070 ± .002	.070 ± .002	.069 ± .002
Corel5k	.127 ± .003	**.059 ± .002**	.149 ± .002	.130 ± .003	.076 ± .004	.069 ± .005	.068 ± .003
ESPgame	.181 ± .002	.120 ± .006	.203 ± .002	.190 ± .002	.142 ± .003	.124 ± .002	**.119 ± .002**
Iaprtc12	.135 ± .002	.089 ± .007	.189 ± .001	.165 ± .002	.106 ± .001	.089 ± .002	**.088 ± .001**
Mirflickr	.224 ± .003	.222 ± .013	.288 ± .001	.285 ± .009	.197 ± .003	.184 ± .003	**.179 ± .003**
Dataset	**SA(↑)**						
	ML*k*NN	SIMM	ICM2L	iMvWL	VLSF	CDMM	MVLD
Emotions	.132 ± .040	.233 ± .032	.106 ± .097	.009 ± .000	.037 ± .017	**.294 ± .043**	.280 ± .037
Yeast	.177 ± .016	.105 ± .019	.004 ± .002	.008 ± .004	.074 ± .009	.207 ± .011	**.222 ± .008**
Pascal07	.028 ± .006	**.394 ± .006**	.038 ± .010	.099 ± .012	.356 ± .003	.359 ± .013	.389 ± .013
Corel5k	.016 ± .002	.032 ± .007	.000 ± .000	.001 ± .001	.034 ± .008	**.069 ± .010**	.046 ± .010
ESPgame	.002 ± .001	.009 ± .001	.000 ± .000	.000 ± .000	.007 ± .002	.013 ± .001	**.013 ± .000**
Iaprtc12	.006 ± .003	.002 ± .001	.000 ± .000	.000 ± .000	.005 ± .001	.018 ± .002	**.026 ± .002**
Mirflickr	**.252 ± .007**	.250 ± .007	.000 ± .000	.000 ± .000	.207 ± .012	.198 ± .009	.203 ± .008



(a) AP. (b) CV. (c) HL. (d) OE. (e) RL. (f) SA.

Fig. 2. The Nemenyi test result of each evaluation metric at the significance level of .05.

generalized to the MVML problem to evaluate the effectiveness of MVLD empirically.

4.1. Data sets

We conduct experiments on several different domain data sets, and Table 2 details the methods we selected for multiple view features of different data, such as image color view, texture view, and shape view in the image data set. For each algorithm, we use five-fold cross-validation for experiments, and they use the same random seed, which is similar to our previous work, so some of the experimental results in this paper are the same. To reduce statistical variability, we averaged ten independent replicates of all experimental results and reported the mean results with standard deviations(std). For the experiments involved in this paper, run under the hardware environment of Windows 10, Intel(R) Core(TM) i7-7700K, 48 GB RAM, and use a consistent random seed to divide the dataset.

4.2. Comparison method

1. ML*k*NN[1]: Concatenate all views to transform the MVML problem into ML learning problem.
2. SIMM[2]: MVML algorithm using shared subspace development and view-specific information extraction. The number of hidden layers is 64. The parameter α is fixed to 1, and β is to searched in the range $\{0.1, 0.01, 0.001, 0.0001\}$.
3. VLSF[3]: An MVML learning algorithm for developing view-label-specific features. The search range of the parameters involved is recommended according to the literature [25].
4. ICM2L[4]: An MVML learning method based on extraction of individuality and common information. The search range of the parameters involved is recommended according to the literature [32].
5. iMvWL[5]: Incomplete multi-View weak-label learning. The complete view information can be obtained in this article. The search range of the parameters involved is recommended according to the literature [30].
6. CDMM[6]: Consistency and diversity neural network MVML learning. The parameters search range is given according to the literature [11].
7. MVLD: MVML learning via view-specific label and label-feature dependence maximization. The search range of non-negative regularization parameters λ_1 and λ_2 are $\{10^{-7}, 10^{-6}, \ldots, 10^{-1}\}$. The search range of non-negative regularization parameters λ_3 is $\{10^{-5}, 10^{-4}, \ldots, 10^{1}\}$. The search range of non-negative regularization parameters λ_4 is $\{10^{3}, 10^{4}, \ldots, 10^{6}\}$. The

1 code: http://palm.seu.edu.cn/zhangml.
2 code: http://palm.seu.edu.cn/zhangml.
3 code: https://jiunhwang.github.io.
4 code: http://www.sdu-idea.cn/codes.php?name=ICM2L.
5 code: http://www.sdu-idea.cn/codes.php?name=iMVWL.
6 code: https://github.com/chengyshaq/CDMM.

Fig. 3. Illustration of the real-time prediction of the MVLD algorithm for some test sample images from the Corel5k dataset. Labels in black fonts represent the labels where the MVLD and ground-truth annotations match each other. Labels in red fonts represent those annotations predicted by MVLD but not present in the corresponding ground truth. Labels in green fonts represent annotations present in the ground truth but not predicted in MVLD.

search range of the kernel function parameter σ is {0.5, 1.0, 5.0, 10}.

For the method using k-NN technology, k is fixed to 10 based on experience. The parameters of all the above techniques are selected through five-fold cross-validation on the training set.

4.3. Evaluation metrics

Evaluation metrics: We measure the performance of each method from six widely used evaluation metrics: Average Precision (AP), Coverage (CV), Hamming Loss (HL), One-Error (OE), Ranking Loss (RL), and Subset Accuracy (SA). The smaller the value of CV, HL, OE, and RL, the better the performance. The larger the value of AP and SA, the better the performance. More detailed information on these evaluation metrics is provided in [1].

4.4. Experimental results and analysis

We report the experimental results of different approaches on 7 MVML data sets in Table 3. The best experimental results are bolded in red, and the second-ranked results are shown in blue.

Friedman test [43] is a widely used strategy for comparing the differences between multiple methods and is usually used for statistical significance analysis. Table 4 summarizes the Friedman statistical F_F values of all evaluation metrics and the critical value at the 0.05 significance level.

Observing Table 4 finds that the F_F statistics are all greater than the critical value, which obviously rejects the null hypothesis that the performance of the comparison methods is equivalent.

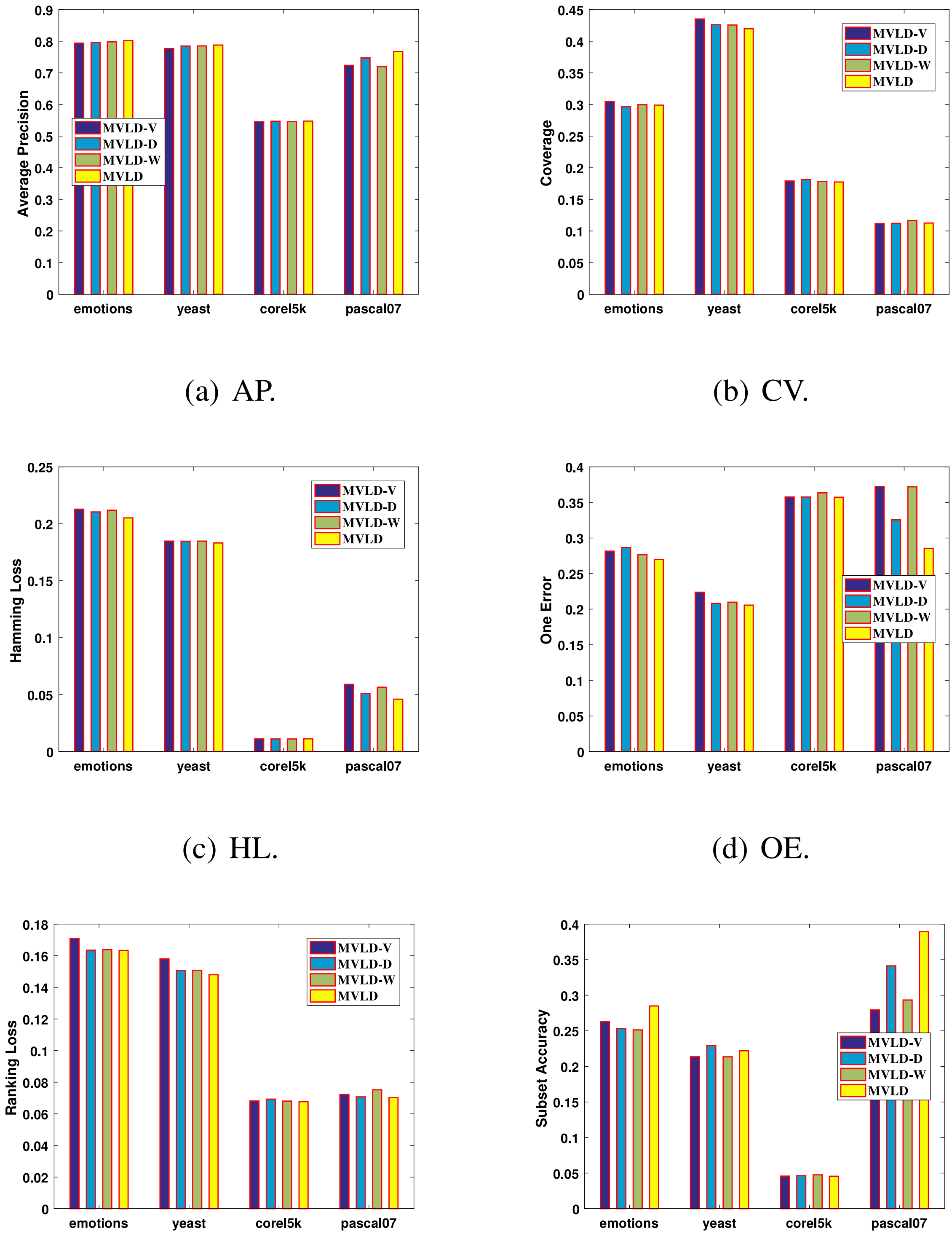
Table 4
Summary of the Friedman statistics F_F ($k = 7$, $N = 7$) of each evaluation metric and the critical value of F(k: Comparing methods; N: Data sets).

Metric	F_F	Critical value ($\alpha = 0.05$)
Average precision	26.6667	
Coverage	27.7377	
Hamming loss	14.3762	2.3638
One-Error	24.0438	
Ranking loss	31.5034	
Subset accuracy	13.6234	

Therefore, we use the *Nemenyi* test [43] as a posterior to illustrate the significant differences between the various methods.

Fig. 2 shows the CD on each evaluation metric among the various comparison methods. For each comparison method, its average ranking is marked as the lower (better) ranking along the axis. If the average ranking difference between the control method and the comparison method is within a CD value, connect them with a solid red line. Based on the experimental results, we have the following conclusions:

1. As shown in Fig. 2, in each evaluation metric, no comparison method has significantly better performance than MVLD. Besides, among the six evaluation metrics, the average ranking of MVLD in the five evaluation metrics is the best, except that it is slightly worse than SIMM in CV. An intuitive reason is that SIMM can leverage the consensus information of all views by utilizing the shared subspace, so SIMM achieves better results on the CV than



(a) AP. (b) CV.

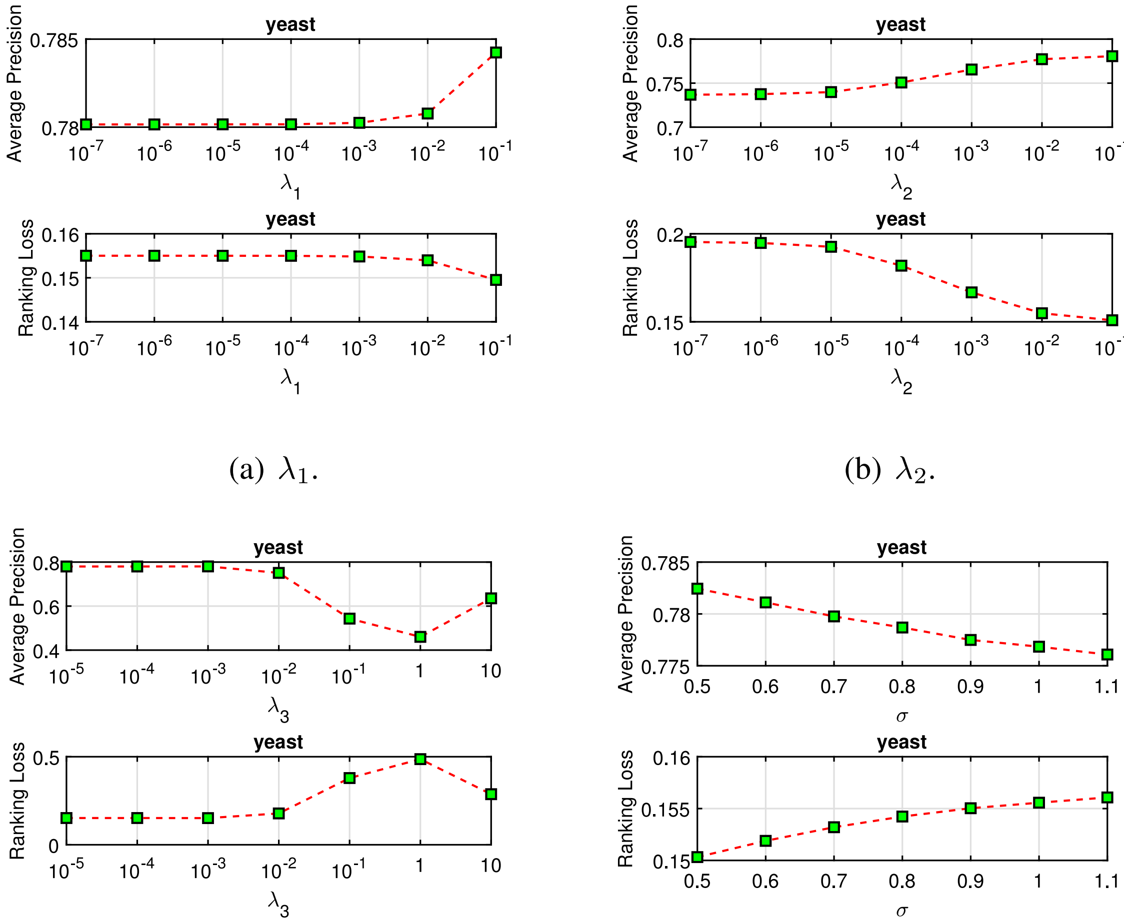
(c) HL. (d) OE.

(e) RL. (f) SA.

Fig. 4. Results of 6 evaluation metrics of MVLD and its variant algorithms on 4 MVML data sets.

MVLD. The use of subspace information can undoubtedly improve the performance of the model, but in the non-aligned view learning problem, the subspace method is no longer applicable because each view cannot communicate information.

2. Compared with CDMM, a kernel function method, MVLD has achieved better performance in each evaluation metric. Compared with linear methods iMvWL, ICM2L, and VLSF, MVLD has excellent performance. Among them, MVLD is significantly better than iMvWL and ICM2L in terms of performance. It can be seen that in the face of heterogeneous data, the general linear method cannot obtain better performance, which verifies the effectiveness of my extension of the kernel method to solve the MVML problem.
3. Compared with the degradation strategy MVML method ML*k*NN, MVLD has achieved excellent performance in every



(a) λ_1. (b) λ_2.

(c) λ_3. (d) σ.

Fig. 5. Parameter sensitivity analysis of MVLD on AP(↑) and RL(↓) based on *Yeast* data set.

evaluation index. It can be seen that the method of directly concatenating will ignore the difference in physical meaning among heterogeneous views and will cause the data dimension to be too large and easily lead to overfitting.

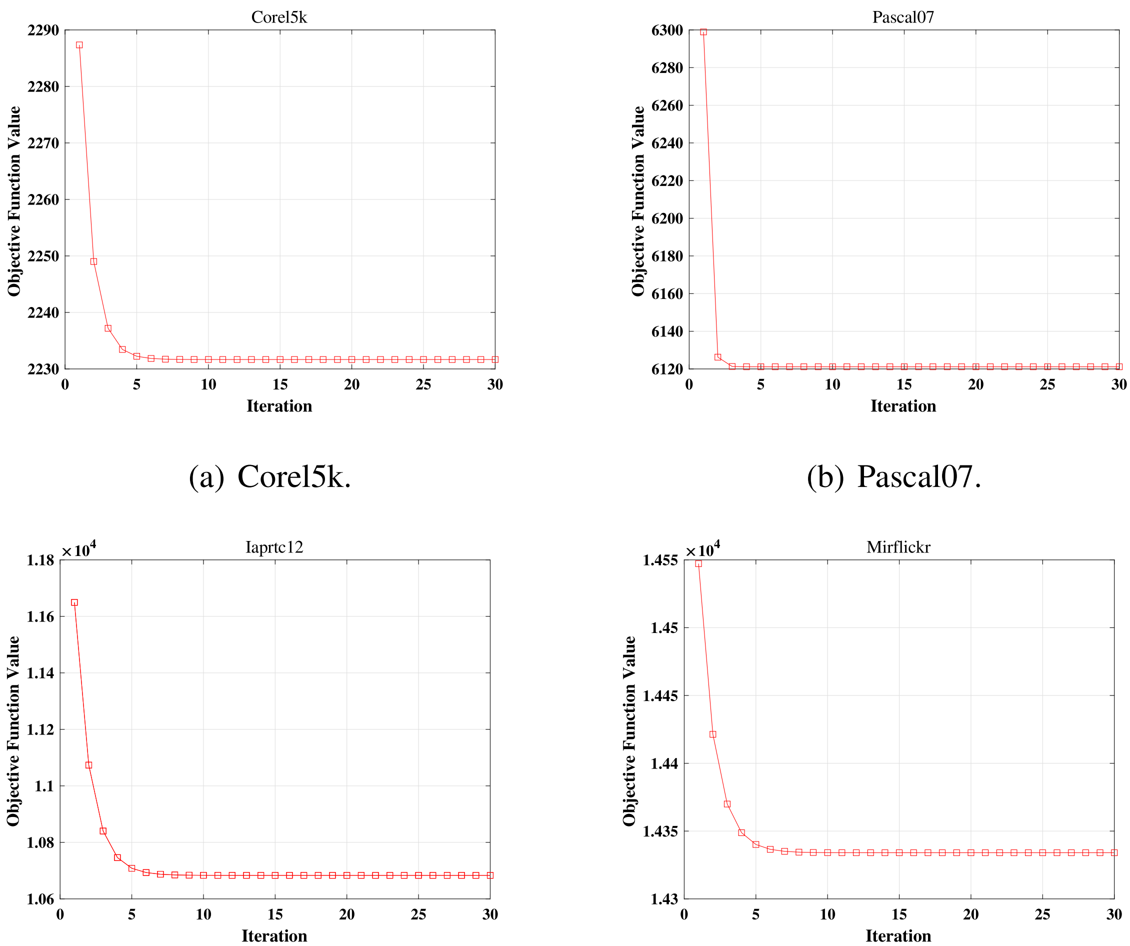
4. As shown in Table 3, the proportion of MVLD ranked first in all results is 54.7%, and the ratio of ranking second is 4.5%. On the one hand, MVLD achieves superior performance on various datasets, verifying that MVLD can be applied to different types of multi-view data. On the other hand, the standard deviation of MVLD in each evaluation metric is also smaller, which shows that it has relatively stable experimental results. It is not difficult to find that MVLD has advantageous robustness in these two aspects.

In Fig. 3, the black font labels represent the labels where the MVLD and ground-truth annotations match each other. Labels in red font represent those annotations predicted by MVLD but not present in the corresponding ground truth annotations. Labels in green font represent labels present in the ground truth but not predicted by MVLD. Observing Fig. 3 shows that MVLD is effective, which can correctly predict the labels of most images. MVLD can learn some unlabeled correct label information in the original ground-truth labels, which is crucial and practical in practical applications because we currently cannot obtain many absolutely accurately labeled images. Additionally, we can also observe the drawback of MVLD, which cannot deal with the problem of class label imbalance very well, resulting in some rare label classes being ignored during prediction, limiting the classification accuracy of MVLD. In future work, we will further study this issue.

4.5. Effectiveness of MVLD via component analysis

To verify the effectiveness of MVLD, we conducted a component analysis of MVLD on the *Emotions*, *Yeast*, *Corel*5*k*, and *Pascal*07 datasets. We mainly analyze the influence of view-specific labels and the interdependence of labels and features on MVLD. To this end, we propose three variants of MVLD: MVLD-V means ignoring the effect of view-specific labels; MVLD-D means ignoring the impact of the interdependence of labels and features; MVLD-W means ignoring the influence of different contributions of views. Fig. 4 shows the experimental results of these three variant algorithms and MVLD on six evaluation metrics.

From the results of each sub-graph in Fig. 4, it can be observed that the results of MVLD on multiple evaluation metrics are better than its different variant algorithms. Specifically, MVLD achieves better results than the MVLD-V model in most cases, which further confirms the effectiveness and feasibility of assigning a specific label to each view for MVML learning. The results compared with MVLD-D also verify that label-feature dependence learning can effectively improve the performance of the classifier. Intuitively, MVLD-V has a better effect than MVLD-D, which shows that the method that considers view-specific feature learning has a more significant contribution to algorithm performance improvement. In general, these results confirm the rationality and



(a) Corel5k. (b) Pascal07.

(c) Iaprtc12. (d) Mirflickr.

Fig. 6. Convergence trend of MVLD.

effectiveness of our model on view-specific feature learning and label-feature dependence and also clarify our motivation.

4.6. Parameter sensitivity analysis

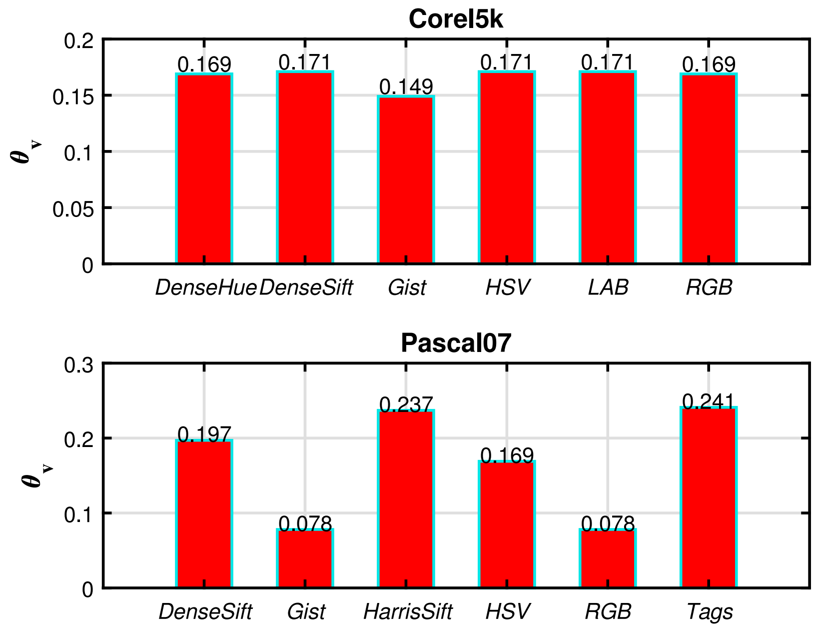
MVLD has 4 main parameters λ_1, λ_2, λ_3, and σ. λ_1 controls the structural differences among views. λ_2 controls the weight coefficient. λ_3 controls label and feature dependencies. σ is the parameter of the RBF kernel function. To verify the sensitivity of the parameter, we tested it on the *Yeast* data set and reported the change results of each parameter on the *AP* and *RL* evaluation metrics in Fig. 5. Specifically, when testing, we change the value of one of the variables within a given interval, while the other variable values are given in advance ($\lambda_1 = 10^{-1}$, $\lambda_2 = 10^{-3}$, $\lambda_3 = 10^{-3}$, $\sigma = 0.5$). For the parameters λ_1 and λ_2 we set the range to $\left\{10^{-7}, 10^{-6}, \ldots, 10^{-1}\right\}$, and for the parameters, λ_3 and σ parameter ranges are set to $\left\{10^{-5}, 10^{-4}, \ldots, 10^{1}\right\}$ and $\{0.5, 0.6, \ldots, 1.1\}$.

Although the value of the regularization parameter is sensitive to MVLD performance, in most cases, more stable results can be obtained so that we can achieve satisfactory results in real applications.

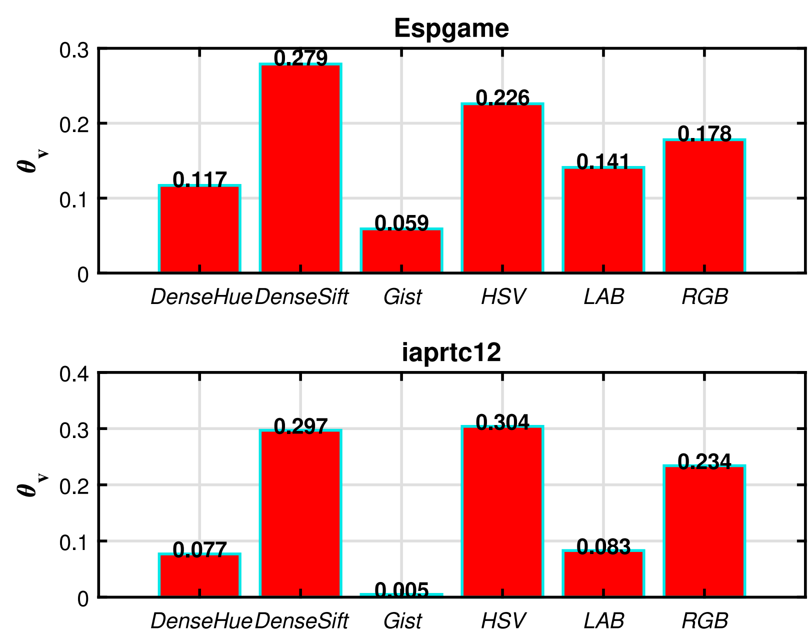
1. From Fig. 5(a), we can see that the performance of MVLD is relatively better when the λ_1 value is around 10^{-1}. And the performance drops significantly from 10^{-1} to 10^{-2}. This result further confirms the role of view-specific labels learning.
2. From Figs. 5(b) and 5(c), we can observe that when the λ_2 and λ_3 values are around 10^{-2}, and 10^{-3}, respectively, the performance of MVLD is relatively stable. Note that when the value of λ_2 continues to increase, the performance of *AP* and *RL* increases, at 10^{-5} it proliferates, and at 10^{-2} it starts to rise slowly. When the value of λ_3 initially increases, the performance is stable, but after 10^{-2}, the performance of MVLD drops sharply, and then there is a small improvement. It can be seen that MVLD is more sensitive to selecting the regularization parameter λ_3. The large performance change also shows that considering the label-feature dependence has a certain significance for improving algorithm performance.
3. We observe Fig. 5(d) and find that as the value of σ increases, the performance decreases, but the overall decrease is not significant.

4.7. Further analysis

To clarify the convergence of the MVLD in the first stage of view-specific label learning, we report the convergence curve of the first stage of it on the *Pascal*07, *Core*15*k*, *Iaprtc*12, and *Mirflickr* datasets in Fig. 6. It can be seen that MVLD tends to converge after 10 iterations in most cases.



(a) Corel5k and Pascal07.



(b) Iaprtc12 and Mirflickr.

Fig. 7. View weights learned by MVLD.

To analyze the impact of the contribution of each view, in Fig. 7, we report the contribution weights of each view on *Core*15*k*, *Pascal*07, *Iaprtc*12 and *Mirflickr* data set(the larger the value, the higher the contribution). Observe Fig. 7 and find that the contribution weights of DenseSift, HSV, and RGB are greater than Gist, LAB, and DenseHue. These comparisons prove the effectiveness of MVLD for ensemble learning in prediction.

5. Conclusion

This paper discussed how to mine the view-specific labels and label-feature dependencies of each heterogeneous view to achieve effective MVML classification. For this reason, we propose a two-stage MVML learning method named MVLD, where the first stage allocates a dedicated label set for each heterogeneous view, and the second stage captures the dependency relationship between labels and features in a principled way integrated classification. First, use the topological structure between different view features to construct a view-specific label set. Then, a multi-label classification model is constructed using the interdependence between labels and features, and the kernel method is utilized to extend the model to a nonlinear approach. Finally, the prediction results of all classifications are integrated with the learned contribution weight to obtain the final prediction results. Experiments on multiple benchmark MVML data sets show that the MVLD model is superior to competing solutions. The experimental results of component analysis of MVLD verify the effectiveness of MVML learning by using view-specific labels and label-feature dependence.

There are two main drawbacks in this paper, and one is that the results of view-specific labeling and classification obtained by the two-stage method we adopt are often sub-optimal. The other is that a single kernel function is used for multi-view kernel mapping, ignoring the problem that the performance of the kernel method is highly related to the selection of the kernel function. In future work, we will study the multi-kernel-based MVML learning problem and are devoted to proposing novel methods for joint learning of view-specific labels and MVML classification models.

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.

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 work is supported by the National Natural Science Foundation of China under Grants 62071001, Nature Science Foundation of Anhui (2008085MF183).

References

[1] Minling Zhang, Zhihua Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng. 26 (8) (2014) 1819–1837.
[2] Grigorios Tsoumakas, Ioannis Katakis, Ioannis Vlahavas, Mining multi-label data, in: Data Mining and Knowledge Discovery Handbook, Springer, 2009, pp. 667–685.
[3] 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.
[4] Ruihuang Li, Changqing Zhang, Huazhu Fu, Xi Peng, Tianyi Zhou, Qinghua Hu, Reciprocal multi-layer subspace learning for multi-view clustering, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 8172–8180.
[5] Tao Zhou, Changqing Zhang, Chen Gong, Harish Bhaskar, Jie Yang, Multiview latent space learning with feature redundancy minimization, IEEE Trans. Cybern. 50 (4) (2018) 1655–1668.
[6] Maria Brbić, Ivica Kopriva, Multi-view low-rank sparse subspace clustering, Pattern Recognit. 73 (2018) 247–258.
[7] Xiaofeng Zhu, Xuelong Li, Shichao Zhang, Block-row sparse multiview multilabel learning for image classification, IEEE Trans. Cybern. 46 (2) (2015) 450–461.
[8] Yongshan Zhang, Jia Wu, Zhihua Cai, S. Yu Philip, Multi-view multi-label learning with sparse feature selection for image annotation, IEEE Trans. Multimed. 22 (11) (2020) 2844–2857.
[9] Jundong Shen, Yi Zhang, Cheng Yu, Chongjun Wang, Multi-view multi-label learning with dual-attention networks for stroke screen, in: 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2020, pp. 1124–1128.
[10] Jing-Han Wu, Xuan Wu, Qing-Guo Chen, Yao Hu, Min-Ling Zhang, Feature-induced manifold disambiguation for multi-view partial multi-label learning, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 557–565.
[11] Dawei Zhao, Qingwei Gao, Yixiang Lu, Dong Sun, Yusheng Cheng, Consistency and diversity neural network multi-view multi-label learning, Knowl.-Based Syst. 218 (2021) 106841.
[12] Aminu Da'u, Naomie Salim, Recommendation system based on deep learning methods: a systematic review and new directions, Artif. Intell. Rev. 53 (4) (2020) 2709–2748.
[13] Xiang Li, Songcan Chen, A concise yet effective model for non-aligned incomplete multi-view and missing multi-label learning, IEEE Trans. Pattern Anal. Mach. Intell. (2021).
[14] Hung-Yi Lo, Ju-Chiang Wang, Hsin-Min Wang, Shou-De Lin, Cost-sensitive multi-label learning for audio tag annotation and retrieval, IEEE Trans. Multimed. 13 (3) (2011) 518–529.
[15] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, Christopher M. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (9) (2004) 1757–1771.
[16] Ling Shao, Fan Zhu, Xuelong Li, Transfer learning for visual categorization: A survey, IEEE Trans. Neural Netw. Learn. Syst. 26 (5) (2014) 1019–1034.
[17] Robert E. Schapire, Yoram Singer, BoosTexter: A boosting-based system for text categorization, Mach. Learn. 39 (2) (2000) 135–168.
[18] Grigorios Tsoumakas, Ioannis Katakis, Ioannis Vlahavas, Random k-labelsets for multilabel classification, IEEE Trans. Knowl. Data Eng. 23 (7) (2010) 1079–1089.
[19] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, Klaus Brinker, Multilabel classification via calibrated label ranking, Mach. Learn. 73 (2) (2008) 133–153.
[20] André Elisseeff, Jason Weston, et al., A kernel method for multi-labelled classification, in: NIPS, Vol. 14, 2001, pp. 681–687.
[21] Min-Ling Zhang, Zhi-Hua Zhou, ML-KNN: A lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048.
[22] Min-Ling Zhang, José M. Peña, Victor Robles, Feature selection for multi-label naive Bayes classification, Inform. Sci. 179 (19) (2009) 3218–3229.
[23] Min-Ling Zhang, Lei Wu, Lift: Multi-label learning with label-specific features, IEEE Trans. Pattern Anal. Mach. Intell. 37 (1) (2014) 107–120.
[24] Jun Huang, Guorong Li, Qingming Huang, Xindong Wu, Learning label specific features for multi-label classification, in: 2015 IEEE International Conference on Data Mining, IEEE, 2015, pp. 181–190.
[25] 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.
[26] Ze-Sen Chen, Xuan Wu, Qing-Guo Chen, Yao Hu, Min-Ling Zhang, Multi-view partial multi-label learning with graph-based disambiguation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 3553–3560.
[27] 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.
[28] 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 AAAI Conference on Artificial Intelligence, Vol. 29, 2015.
[29] Fangwen Zhang, Xiuyi Jia, Weiwei Li, Tensor-based multi-view label enhancement for multi-label learning, in: IJCAI, 2020, pp. 2369–2375.
[30] Qiaoyu Tan, Guoxian Yu, Carlotta Domeniconi, Jun Wang, Zili Zhang, Incomplete multi-view weak-label learning, in: IJCAI, 2018, pp. 2703–2709.
[31] Changqing Zhang, Ziwei Yu, Qinghua Hu, Pengfei Zhu, Xinwang Liu, Xiaobo Wang, Latent semantic aware multi-view multi-label classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.
[32] Qiaoyu Tan, Guoxian Yu, Jun Wang, Carlotta Domeniconi, Xiangliang Zhang, Individuality-and commonality-based multiview multilabel learning, IEEE Trans. Cybern. 51 (3) (2021) 1716–1727.
[33] Zheng Fang, Zhongfei Zhang, Simultaneously combining multi-view multi-label learning with maximum margin classification, in: 2012 IEEE 12th International Conference on Data Mining, IEEE, 2012, pp. 864–869.
[34] Dawei Zhao, Qingwei Gao, Yixiang Lu, Dong Sun, Two-step multi-view and multi-label learning with missing label via subspace learning, Appl. Soft Comput. 102 (2021) 107120.
[35] Pengfei Zhu, Qi Hu, Qinghua Hu, Changqing Zhang, Zhizhao Feng, Multi-view label embedding, Pattern Recognit. 84 (2018) 126–135.
[36] Xuan Wu, Qing-Guo Chen, Yao Hu, Dengbao Wang, Xiaodong Chang, Xiaobo Wang, Min-Ling Zhang, Multi-view multi-label learning with view-specific information extraction, in: IJCAI, 2019, pp. 3884–3890.

[37] 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.
[38] Gaston K. Mazandu, Emile R. Chimusa, Nicola J. Mulder, Gene ontology semantic similarity tools: survey on features and challenges for biological knowledge discovery, Brief. Bioinform. 18 (5) (2017) 886–901.
[39] Lei Feng, Bo An, Partial label learning with self-guided retraining, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 3542–3549.
[40] Lei Feng, Jun Huang, Senlin Shu, Bo An, Regularized matrix factorization for multilabel learning with missing labels, IEEE Trans. Cybern. (2020) 1–12.
[41] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, 2002.
[42] Yong Luo, Dacheng Tao, Chang Xu, Chao Xu, Hong Liu, Yonggang Wen, Multiview vector-valued manifold regularization for multilabel image classification, IEEE Trans. Neural Netw. Learn. Syst. 24 (5) (2013) 709–722.
[43] Janez Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.