A Multi-View Learning Framework with A Linear Computational Cost

Xiaowei Xue, Feiping Nie*, Zhihui Li, Sen Wang, Xue Li, and Min Yao

***Abstract*—Learning features from multiple views has attracted much research attention in different machine learning tasks, such as multi-class and multi-label classification problems. In this paper, we propose a Multi-class Multi-label Multi-view learning framework with a linear computational cost where an example is associated with at least one label and represented by multiple information sources. We simultaneously analyze all features by learning an integrated projection matrix. We can also automatically select more important views for subsequent classifier to predict each class. As the proposed objective function is non-smooth and difficult to solve, we apply a novel optimization method that converts the multi-view learning problem to a set of linear single-view learning problems by bridging our problem to an easily solvable approach. Compared to the conventional methods which learn the entire projection matrix, our algorithm independently optimizes each column of the projection matrix for each class, which can be easily parallelized. In each column optimization, the most computationally intensive step is pure and simple matrix-by-vector multiplication. As a result, our algorithm is much more applicable to large-scale problems than the multi-view learning methods with a non-linear computational cost. Moreover, rigorous convergence proof of the proposed algorithm is also provided. To evaluate the effectiveness of the proposed approach, experimental comparisons are made with state-of-the-art algorithms in multi-class and multi-label classification tasks on many multi-view benchmarks. We also report the efficiency comparison results on different numbers of data samples. The experimental results demonstrate that our algorithm can achieve superior performance to all the compared algorithms.**

***Index Terms*—Multi-view learning, primal SVM, hinge loss, a linear computational cost.**

I. Introduction

In traditional multi-class classification problems, they classify samples into one of the more than two classes. However, in the real world, an object may be very complicated and have multiple semantic meanings. For example, in image annotation applications, an image may be related to several objects, such as birds, trees and rivers; in text category, news may belong to the political topic and legal topic simultaneously. Based on above consideration, the paradigm of multi-label learning naturally emerges and extends multi-class classification by allowing a sample to be associated with multiple labels.

Usually, the traditional multi-class or multi-label methods can just deal with a single type of features. However, in recent years, data representation is becoming more diverse than before. In many real applications, the datasets are described in the form of multiple views by being collected from different sources or obtained in various feature construction ways. For instance, the multimedia data can be described by text, video, image and audio components. On the other hand, the features for visual objects can be extracted by Histogram of Oriented Gradients (HOG)[1], Speeded-Up Robust Features (SURF) [2] and Scale-Invariant Feature Transform (SIFT) [3]. It is intuitive that aggregating more data from different views can yield a more informative description of an object than only using a single source. For the conventional single-view multi-label algorithms [4], [5], [6], it is difficult to directly handle such problems effectively where samples are represented by multi-view features. The straightforward way to utilize heterogeneous features is to form a longer feature vector by concatenating all of them into one single view. Obviously, the correlation among views is little considered and the curse of dimensionality will be detrimental to real-world applications. Besides, the feature concatenation often leads to a huge matrix to be completed, which would make the time cost very high and sometimes intolerable. In contrast to the traditional single-view learning, multi-view learning methods [7], [8] aim to fuse different views to bring in more information rather than using single view and improve the overall learning performance [9], [10], [11].

To tackle the multi-class multi-label multi-view learning problems, a number of sophisticated approaches have been proposed to make full use of the benefit of multi-view features. One solution, Multiple Kernel Learning (MKL) [12], adopts either linear or non-linear combination of multiple kernels to integrate the heterogeneous features from different views for multi-class or multi-label learning. In [13], Ji et al. adopted multiple kernel learning with the help of a hypergraph to capture the correlation information for multi-label learning. [12] combined the vector-valued function, manifold regularization and MKL to handle multi-label problems. However, as pointed out in [14], the computational complexity of MKL mainly depends on the training methods, and most of the existing MKL algorithms [15], [16] have computational burden when dealing with high-dimensional large-scale datasets. Another popular solution, subspace learning [17], exploits a latent lower-dimensional subspace shared by different views. Canon-

Corresponding author: Feiping Nie.

Xiaowei Xue and Min Yao are with the College of Computer Science, Zhejiang University, Hangzhou 310027, China (Email:xwxue@zju.edu.cn; myao@zju.edu.cn)

Feiping Nie (corresponding author) is with the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China (Email: feipingnie@gmail.com)

Zhihui Li is with Beijing Etrol Technologies Co., Ltd. (Email: zhihuilics@gmail.com)

Sen Wang is with School of Information and Communication Technology, Griffith University, Australia (Email: sen.wang@griffith.edu.au)

Xue li is with the School of Information Technology and Electrical Engineering, The University of Queensland, Australia (Email:xueli@itee.uq.edu.au)

ical Correlation Analysis (CCA) and its kernel version, namely KCCA [18], are representative subspace learning algorithms in which the correlations among views are discovered in a shared space and then utilized to improve the performance of the subsequent learning tasks. Inspired by fisher criterion in single-view learning, many works [19], [20] have applied fisher discriminant analysis in multiple views to seek informative projections with the help of label information. Besides, it is not accurate to assume features of all the views are equally important for each class. To solve this problem, some researcher adopted regularization-based multi-view learning techniques. For example, $l_{2,1}$-norm [21] has been applied to solve clustering problems by learning weights for each feature on each cluster individually [22]. However, integrating different norms into models always adds models' computational complexities, which is not applicable to the large-scale datasets.

Although the aforementioned methods have achieved satisfactory performance on small-scale problems, their capability of dealing with large-scale multi-view datasets is often limited by the computational complexity. To address this drawback, in this paper, we propose a linear and convex multi-view learning method that is suitable for both multi-class and multi-label classification problems. The proposed method is mainly inspired by Support Vector Machine (SVM), and the hinge loss function [23] is adopted as the hinge loss is usually better than the Least Square loss as well as logistic loss in term of classification tasks [24]. In our method, data in each view is separately learned with labels and then combined with weights of different views. Similar to the works in [25], [26], each view has a weight to measure their importance for the classifier to predict each class. Through optimizing these weights and determining their values, our model can automatically select more important views for predicting each class. Simultaneously, our algorithm optimizes the total class margins across all views. Instead of directly optimizing the non-smooth objective function, we propose an efficient optimization method to convert the multi-view problem into a set of linear single-view learning problems by bridging the new problem with an easily solvable and efficient optimization problem for multi-class multi-label multi-view tasks. Specifically, different from traditional multi-view learning algorithms that entirely learn the weights of multiple views, our method individually optimizes each column of the projection matrix for each class rather than learning the entire feature weight matrix. It is worth noting that the most computational step in our algorithm is a set of matrix-by-vector multiplications instead of the matrix-by-matrix multiplication of the entire weight matrix. Our algorithm can be easily parallelized to achieve much higher concurrency than the peers who analyze the entire matrix. Moreover, the convergence of the proposed algorithm is guaranteed by a rigorous theoretical proof. As a result, our method is much more applicable to large-scale multi-view problems than most of the existing counterparts. Experimental results demonstrate that our algorithm is superior to all the compared algorithms in both multi-view multi-class and multi-view multi-label classification tasks. Rapid convergence to the global optima and linear running cost makes our algorithm further stand out against all the compared algorithms. We name our proposed algorithm Linear Multi-View Learning Framework (LMVL). To summarize, the major contributions of this work are summarized as follows:

- We propose a novel multi-view learning algorithm that simultaneously learns all the features from multiple views and learns a weight for each view to measure its importance for the subsequent classifier to predict each class. The proposed algorithm is suitable for both multi-view multi-class and multi-view multi-label classification problems.
- We can automatically select more important views for predicting each class.
- To optimize the objective function, we propose an efficient algorithm with guaranteed convergence by converting the original multi-view problem into a set of linearly solvable single-view problem, which can be easily implemented in parallel on a multi-core machine.
- Extensive experiments are conducted on several benchmark datasets for multi-class and multi-label classification tasks. The experimental results demonstrate that our method consistently achieves better performance than state-of-the-art multi-view methods. In addition, computational complexity comparison shows that our model as well as its parallelized version can deal with large-scale multi-view datasets while preserving decent performance.

II.Related Work

Our work is mainly related to two machine learning topics: multi-label learning and multi-view learning. Here we review some related works in these two areas.

A. Multi-Label Learning

Multi-label learning is motivated by the fact that in real world objects naturally involve multiple attributes. It studies the problems where each sample is associated with multiple labels. During the past decades, a number of multi-label methods were proposed and widely applied in many real applications. Some of these methods transform multi-label learning problems into other well-established learning scenarios. For example, [27] decomposed the multi-label problems into several independent binary classification problems. Different from using binary classifiers, [4] converted the multi-label learning into label ranking problems with the help of pairwise comparison techniques to fulfill the ranking among labels.

On the other hand, some works concentrate on extending popular learning techniques to deal with multi-label learning problems. Multiple k nearest neighbours (ML-KNN) [5] extended the KNN method and utilized maximum a posteriori rule to predict by reasoning with the labeling information of neighbors for multi-label problems. A ranking method was presented in [28] with maximum margin strategy to minimize the ranking loss function of SVM. In [29], the decision tree is recursively built by adopting multi-label entropy as information gain criterion to deal with multi-label data.

However, none of the methods mentioned above considers the properties of heterogeneous features from different views.

For some complicated tasks such as image classification, one single kind of feature always cannot describe the objects very well. For example, the HOG mentioned above is used to describe the shape information of images while SIFT is robust to image noise, illumination changes and rotation. Therefore, how to integrate the heterogeneous features properly with well-designed machine learning algorithm is very important for many multi-label applications.

B. Multi-View Learning

Multi-view learning deals with the data described in multiple views and its goal is to exploit the relations among different views to improve the final performance. It has attracted more and more research attention over the last decade and has been well studied and applied to a number of applications of data mining [30], computer vision [31], [32], and machine learning [33], [34]. In this section, we review the multi-view learning from the perspective of feature fusion and mainly focus on the classification methods. According to the level of feature fusion, the multi-view classification methods can be roughly categorized into tow groups [35]: 1) feature-level fusion, 2) classifier-level fusion.

1) Feature-level fusion. In the feature-level multi-view classification methods, the representative works are Multiple Kernel Learning (MKL) and subspace learning. In MKL, each kernel can be regarded as a view. A typical MKL algorithm aims to learn an ensemble of multiple kernels for better performance of a certain application rather than just using a single kernel. In [36], Lanckriet et al. solved MKL with semidefinite programming techniques for binary classification. To improve the efficiency of MKL, Sonnenburg et al.[37] adopted the semi-infinite linear program to optimize MKL for large scale data. However, both of these MKL methods were based on SVM for binary classification problems and are not naturally designed for multi-class or multi-label classification problems. Some other MKL methods try to introduce various regularizers to learn an appropriate kernel combination, including l_1-norm [38], l_2-norm [15], l_∞-norm [38] and so on. However, these MKL methods with different norms are constrained to small datasets or a limited number of base kernels, making it difficult to solve large-scale datasets in real applications.

Different from MKL learning, another kind of approaches in feature-level fusion is multi-view subspace learning, where common latent representation is tried to be extracted. Canonical correlation analysis (CCA) [18] is one of the most popular subspace learning methods that can be used to find a linear mapping that maximizes the cross-correlation between two views. [18] extended the conventional CCA to sparse kernel CCA with kernel tricks and l_1-norm. In [39], White et al. provided a convex formulation for multi-view subspace learning to learn a low dimensional representation. To model the correlations between different views, [17] used the Gaussian process regression to learn common hidden structure shared among views while Memisevic et al. [40] constructed the joint embedding for two views to find a low-dimensional latent distribution by maximizing mutual information.

2) Classifier-level fusion. In the classifier-level fusion, the straightforward way is to fuse the decisions made by different classifiers learned from different independent view [41]. As shown in [42], Fumera et al. presented a theoretical and experimental analysis of linear combiners for multiple classifier systems and demonstrated its effectiveness. In contrast to simple linear fusion, in [43], Snoek et al. adopted a probabilistic aggregation mechanism to fuse different SVM outputs. Specifically, each view is independently used to train an SVM learner and then the output of each SVM is converted to probabilistic scores. Finally, all the scores are concatenated as the input of another SVM for final classification.

Another kind of classifier-level fusion methods improves the final performance through cooperation between different views and always belongs to semi-supervised learning[44], such as co-training. As one of the representative works on semi-supervised multi-view learning, Co-training was first introduced in [45] and trained alternately to maximize the mutual agreement on two distinct views of the unlabeled data. Assuming that each data point is described by independent features in two views, co-training trains a classifier using labeled data in each view. Predictions on new unlabeled data in one view are mutually used to enlarge the training set of the other view. Other learning techniques have also been combined to achieve better learning results in different applications. Expectation-maximization has been combined with co-training in [46] for lower errors. In [47], SVM was used to develop an extended version of co-EM for multi-view learning. Muslea et al. [48] claimed that active learning is beneficial to the co-training regarding robustness in multi-view learning problems. The conditional independent assumption is important for co-training theoretically and empirically [35], but it rarely holds in real-world applications.

III. A MULTI-VIEW LEARNING METHOD WITH A LINEAR COMPUTATIONAL COST FOR MULTI-CLASS AND MULTI-LABEL CLASSIFICATION TASKS

In this section, we will first systematically propose a novel multi-view learning model for multi-class and multi-label classification tasks. Then a new efficient iterative algorithm is introduced to solve the highly non-smooth objective by converting the multi-view problem into a set of linearly solvable single-view problem.

Before going into the details of our algorithm, let us introduce some notations. In this paper, we write the matrices as bold uppercase letter and vector as bold lowercase letters. Given a set of n data samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$, the data matrix in the j-th view is $\mathbf{X}^{(j)} = [\mathbf{x}_1^{(j)}, \cdots, \mathbf{x}_n^{(j)}] \in \mathbb{R}^{d_j \times n} (j = 1, \cdots, v)$, where v is the number of views and d_j denotes the feature dimension of the j-th view. $\mathbf{y}_i \in \mathbb{R}^c$ is the class label vector of sample $\mathbf{x}_i$, where c is the number of classes. If $\mathbf{x}_i$ belongs to the class k, then $y_i^k = 1$, otherwise $y_i^k = -1$. Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ with $d = \sum_{j=1}^{v} d_j$ and $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_n] \in \mathbb{R}^{c \times n}$.

A. The Proposed Framework

In our method, a feature weight matrix $\mathbf{W} = [\mathbf{w}_1^1, \cdots, \mathbf{w}_1^c; \cdots; \cdots; \cdots; \mathbf{w}_v^1, \cdots, \mathbf{w}_v^c] \in \mathbb{R}^{d\times c}$ is directly learned, where $\mathbf{w}_q^p \in \mathbb{R}^{d_q}$ indicates the weights of all the features from the q-th view in the classification decision function of the p-th class. Typically, we adopt a convex loss function $\mathcal{L}(\mathbf{X}, \mathbf{W})$ to measure the loss incurred by $\mathbf{W}$ for all the training samples. In this paper, we utilize the hinge loss since hinge loss based Support Vector Machine (SVM) usually achieves better performance in terms of classification than least square loss or logistic loss [50].

Our goal is to classify each sample into c classes by exploiting the interrelationship of all the features from different views and selecting more important views for the subsequent classifier to predict each class. In our method, LMVL learns c linear functions with parameters $\{\mathbf{w}_k, b_k\}_{k=1}^c$. For multi-class and multi-label classification problems, we adopt different mechanisms. The original SVM was proposed for solving binary classification problems. Although it was extended to solve multi-class classification problems via one versus one (OVO) or one versus all (OVA) strategy by simply breaking the multi-class problems into several binary classification problems, it may not be efficient as several SVM classifiers need to be trained. Moreover, it ignores the correlation between classes. To overcome the drawbacks of binary SVM, we consider all classes at once by solving only one single optimization problem for both multi-class and multi-label classification problems. For multi-class classification problems, we adopt the OVA strategy while for multi-label classification problems, we transform it into a set of independent binary classification problems for each class via the one-vs-others scheme which is a conceptually simple and computationally efficient solution.

In our method, the decision function for the i-th sample is $\arg\max_{1\le k\le c}(\mathbf{w}^k)_k^T\mathbf{x}_i + b_k$. Here, we propose the method by minimizing the following objective function:

$$\min_{\mathbf{W},\mathbf{b}} \frac{1}{2}\|\mathbf{W}\|_F^2 + C\sum_{k=1}^{c}\sum_{i=1}^{n}(1 - \mathbf{y}_i^k(\sum_{j=1}^{v}(\sqrt{\theta_j^k}\mathbf{w}_j^k)^T\mathbf{x}_i^{(j)} + b_k))_+ \tag{1}$$

where $\sum_{j=1}^{v}\theta_j^k = 1$, $\theta_j^k \ge 0$, $C > 0$ is a trade-off parameter, the function $(a)_+$ is defined as $(a)_+ = max(0, a)$. According to the Frobenius norm definition, $\|\mathbf{W}\|_F^2 = \sum_{k=1}^{c}\sum_{j=1}^{v}\|\mathbf{w}_j^k\|_2^2$. In Eq. (1), $\sqrt{\theta_j^k}$ is the weight to measure the importance of the j-th view for predicting the k-th class. Through optimizing $\sqrt{\theta_j^k}$ and determine its value, we can automatically select more important views for subsequent classifier to predict each class. Note that the reason why we adopt $\sqrt{\theta_j^k}$ rather than other forms is for the optimization convenience and the details are as follows.

B. Optimization Algorithm

The objective function in Eq. (1) is a highly non-smooth problem and difficult to solve. However, we can find the globally optimal solutions based on the following theorem:

Theorem 1: Minimizing Eq. (1) equals to minimize the following equation:

$$\min_{\mathbf{W},\mathbf{b}} \frac{1}{2}\sum_{k=1}^{c}\left(\sum_{j=1}^{v}\|\mathbf{w}_j^k\|_2\right)^2 + C\sum_{k=1}^{c}\sum_{i=1}^{n}(1 - \mathbf{y}_i^k(\sum_{j=1}^{v}(\mathbf{w}_j^k)^T\mathbf{x}_i^{(j)} + b_k))_+ \tag{2}$$

Proof: Obviously, substituting $\tilde{\mathbf{w}}_j^k$ for $\sqrt{\theta_j^k}\mathbf{w}_j^k$, we can rewrite Eq. (1) as following:

$$\min_{\tilde{\mathbf{W}},\mathbf{b}} \frac{1}{2}\sum_{k=1}^{c}\sum_{j=1}^{v}\frac{\|\tilde{\mathbf{w}}_j^k\|_2^2}{\theta_j^k} + C\sum_{k=1}^{c}\sum_{i=1}^{n}(1 - \mathbf{y}_i^k(\sum_{j=1}^{v}(\tilde{\mathbf{w}}_j^k)^T\mathbf{x}_i^{(j)} + b_k))_+ \tag{3}$$

where $\sum_{j=1}^{v}\theta_j^k = 1$, $\theta_j^k \ge 0$.

After the substitution, the parameter θ_j^k just appears in the first part of Eq. (3). We directly set the derivative of Eq. (3) w.r.t θ_j^k to zero. When $\theta_j^k = \frac{\|\tilde{w}_j^k\|_2}{\sum_{j=1}^{v}\|\tilde{w}_j^k\|_2}$, we can get the minimum value of the first part of Eq. (3) as $\sum_{k=1}^{c}\left(\sum_{j=1}^{v}\|\tilde{\mathbf{w}}_j^k\|_2\right)^2$. In this case, we have:

$$\min_{\tilde{\mathbf{W}}}\sum_{k=1}^{c}\left(\sum_{j=1}^{v}\|\tilde{\mathbf{w}}_j^k\|_2\right)^2 \Leftrightarrow \min_{\tilde{\mathbf{W}},\sum_{j=1}^{v}\theta_j^k=1,\theta_j^k\ge 0}\sum_{k=1}^{c}\sum_{j=1}^{v}\frac{\|\tilde{\mathbf{w}}_j^k\|_2^2}{\theta_j^k} \tag{4}$$

According to Eq. (3) and (4), we have

$$\begin{aligned}
&\min_{\tilde{\mathbf{W}},\mathbf{b}} \frac{1}{2}\sum_{k=1}^{c}\sum_{j=1}^{v}\frac{\|\tilde{\mathbf{w}}_j^k\|_2^2}{\theta_j^k} \\
&\quad + C\sum_{k=1}^{c}\sum_{i=1}^{n}(1 - \mathbf{y}_i^k(\sum_{j=1}^{v}(\tilde{\mathbf{w}}_j^k)^T\mathbf{x}_i^{(j)} + b_k))_+ \\
\Leftrightarrow &\min_{\tilde{\mathbf{W}},\mathbf{b}} \frac{1}{2}\sum_{k=1}^{c}\left(\sum_{j=1}^{v}\|\tilde{\mathbf{w}}_j^k\|_2\right)^2 \\
&\quad + C\sum_{k=1}^{c}\sum_{i=1}^{n}(1 - \mathbf{y}_i^k(\sum_{j=1}^{v}(\tilde{\mathbf{w}}_j^k)^T\mathbf{x}_i^{(j)} + b_k))_+
\end{aligned} \tag{5}$$

Therefore, minimizing the objective function in Eq. (1) equals to solve the following convex problem:

$$\begin{aligned}
&\min_{\mathbf{W},\mathbf{b}} \frac{1}{2}\sum_{k=1}^{c}\left(\sum_{j=1}^{v}\|\mathbf{w}_j^k\|_2\right)^2 \\
&\quad + C\sum_{k=1}^{c}\sum_{i=1}^{n}(1 - \mathbf{y}_i^k(\sum_{j=1}^{v}(\mathbf{w}_j^k)^T\mathbf{x}_i^{(j)} + b_k))_+
\end{aligned} \tag{6}$$

Rather than directly solving the above-mentioned convex problem, we bridge it with a solvable method which has a linear computational cost. By doing so, it makes our method applicable to large-scale problems. In [51], the authors have solved the l_2-norm regularized l_1-norm loss primal SVM with a linear computational cost. In other words, given $\mathbf{X}$ and $\mathbf{Y}$, we have a function to obtain $\mathbf{w}^*$ and b^*:

$$[\mathbf{w}^*, b^*] = \arg\min_{\mathbf{w},b} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))_+ \quad (7)$$

In order to bridge our method with Eq. (7), we individually solve Eq. (6) for each class. In this case, for the k-th class we need to optimize the following problem:

$$\min_{\mathbf{W}_k,\mathbf{b}_k} \frac{1}{2}\left(\sum_{j=1}^{v}\|\mathbf{w}_j^k\|_2\right)^2 \quad +C\sum_{i=1}^{n}(1 - \mathbf{y}_i^k(\sum_{j=1}^{v}(\mathbf{w}_j^k)^T\mathbf{x}_i^{(j)} + b_k))_+ \quad (8)$$

where $\mathbf{W}_k = [\mathbf{w}_1^k; \cdots ; \mathbf{w}_v^k]$ and $\mathbf{W}_k$ is the concatenation of $\mathbf{w}_j^k$. For brevity, we denote $f(\sum\limits_{j=1}^{v}(\mathbf{w}_j^k)^T\mathbf{x}_i^{(j)}, b_k) = (1 - \mathbf{y}_i^k(\sum\limits_{j=1}^{v}(\mathbf{w}_j^k)^T\mathbf{x}_i^{(j)} + b_k))_+$.

Let $\mathcal{J}(\mathbf{W}_k, b_k) = \min\frac{1}{2}\left(\sum\limits_{j=1}^{v}\|\mathbf{w}_j^k\|_2\right)^2 + C\sum\limits_{i=1}^{n}f(\sum\limits_{j=1}^{v}\mathbf{w}_j^k\mathbf{x}_i^{(j)}, b_k)$. By setting the derivative of $\mathcal{J}(\mathbf{W}_k, b_k)$ w.r.t $\mathbf{W}_k$:

$$\mathbf{D}\mathbf{W}_k + C\frac{\partial\sum\limits_{i=1}^{n}f(\sum\limits_{j=1}^{v}(\mathbf{w}_j^k)^T\mathbf{x}_i^{(j)}, b_k)}{\partial\mathbf{W}_k} = 0 \quad (9)$$

where

$$\mathbf{D} = \begin{bmatrix}\mathbf{D}_1 & & \\ & \cdots & \\ & & \mathbf{D}_v\end{bmatrix}_{d\times d}, \mathbf{D}_j = \frac{\sum\limits_{j=1}^{v}\|\mathbf{w}_j^k\|_2}{\|\mathbf{w}_j^k\|_2}\mathbf{I} \in \mathbb{R}^{d_j\times d_j}. \quad (10)$$

Then taking derivative of $\mathcal{J}(\mathbf{W}_k, b_k)$ w.r.t b_k, we can get:

$$\frac{\partial\sum\limits_{i=1}^{n}f(\sum\limits_{j=1}^{v}(\mathbf{w}_j^k)^T\mathbf{x}_i^{(j)}, b_k)}{\partial b_k} = 0 \quad (11)$$

If $\mathbf{D}$ is constant, the optimal solution to the following problem will satisfy Eq. (9) and Eq. (11). $\mathbf{D}$ is updating according to the $\mathbf{W}_k$. We need to minimize the following function:

$$\min_{\mathbf{W}_k,\mathbf{b}_k} \mathbf{W}_k^T\mathbf{D}\mathbf{W}_k + C\sum_{i=1}^{n}f(\sum_{j=1}^{v}(\mathbf{w}_j^k)^T\mathbf{x}_i^{(j)}, b_k) \quad (12)$$

Let $\tilde{\mathbf{W}}_k = \mathbf{D}^{\frac{1}{2}}\mathbf{W}_k$, $\tilde{\mathbf{x}}_i^{(j)} = \left(\frac{\sum\limits_{j=1}^{v}\|\mathbf{w}_j^k\|_2}{\|\mathbf{w}_j^k\|_2}\right)^{-\frac{1}{2}}\mathbf{x}_i^{(j)}$ and $\tilde{\mathbf{w}}_j^k = \left(\frac{\sum\limits_{j=1}^{v}\|\mathbf{w}_j^k\|_2}{\|\mathbf{w}_j^k\|_2}\right)^{\frac{1}{2}}\mathbf{w}_j^k$. Then the problem becomes as following:

$$\min_{\tilde{\mathbf{W}}_k,\mathbf{b}_k} \tilde{\mathbf{W}}_k^T\tilde{\mathbf{W}}_k + C\sum_{i=1}^{n}f(\sum_{j=1}^{v}(\tilde{\mathbf{w}}_j^k)^T\tilde{\mathbf{x}}_i^{(j)}, b_k) \quad (13)$$

So far we have bridged the objective function in Eq. (8) with the solvable objective function in Eq. (7). As we use Eq. (7) (a linear SVM) to optimize each column of the weight matrix $\mathbf{W}$, thus the computational cost of our algorithm is also linear w.r.t. the number of data. We summarize the new proposed method in Algorithm 1.

Algorithm 1: An efficient iterative algorithm to solve the optimization problem in Eq. (6)

```
Input: data $\mathbf{X} \in \mathbb{R}^{d\times n}$, label $\mathbf{Y} \in \mathbb{R}^{c\times n}$, penalty parameter scalar $C$.
Procedure:
1: Initialize the projection matrix
   $\mathbf{W}^{(0)} = \{w_i^j = 1\}(i = 1, ..., d, j = 1, ..., c)$, and
   $b_k^{(o)}|_{k=1}^{c} = 1$.
2: for the $k$-th class, $k = 1, ..., v$
3:     Calculate the diagonal matrix $\mathbf{D}$ using Eq. (10)
4:     Update $\tilde{\mathbf{X}}^{(j)} = \left(\frac{\sum_{j=1}^{v}\|\mathbf{w}_j^k\|_2}{\|\mathbf{w}_j^k\|_2}\right)^{-\frac{1}{2}}\mathbf{X}^{(j)}$
       $\tilde{\mathbf{w}}_j^k = \left(\frac{\sum_{j=1}^{v}\|\mathbf{w}_j^k\|_2}{\|\mathbf{w}_j^k\|_2}\right)^{\frac{1}{2}}\mathbf{w}_j^k$
5:     Calculate $\tilde{\mathbf{W}}_k^{(t+1)}$ and $b_k^{(k+1)}$ using Eq. (7) by
   the algorithm SVM-ALM for $L_p$-primal SVM in
   [51].
6:     update $\mathbf{W}_k = \mathbf{D}^{\frac{1}{2}}\tilde{\mathbf{W}}_k$
7:     Iterate 3-6 until convergence
8: end for
Output: $\mathbf{W} \in \mathbb{R}^{d\times c}$, $b_k|_{k=1}^{c}$.
```

V. Experiment

In this section, we experimentally evaluate the performance of the proposed approach in both multi-class and multi-label classification tasks. The experiments are divided into two parts. The first part is to demonstrate the superiority of LMVL for multi-view multi-class classification tasks while the second part shows the effectiveness of LMVL compared to state-of-the-art methods for multi-view multi-label classification problems. We also report the running time and exam the convergence of our method.

A. Compared Algorithms

To evaluate the performance of our method in multi-view multi-class tasks and multi-view multi-label classification tasks, we compared the proposed LMVL with several single-view and multi-view algorithms. Brief descriptions of these compared algorithms are given as follows:

1) SVM: We use the standard SVM with each type of features and the concatenation of all the features. In our experiments, all the SVM is implemented by LIBSVM software package [52]. In the running time analysis, we adopt linear SVM. The computational complexity of standard SVM is $O(dn^2)$.
2) SVM l_p MKL [38]: A popular SVM-based MKL algorithm extends MKL with different norms. According to [38], different kernel normalizations can have a significant impact on the performance of MKL. In our experiments, we adopt the popular l_1, l_2 and l_∞ norms to normalize the MKL methods. The computational complexities of SVM l_∞ MKL, SVM l_1 MKL and

SVM l_2 MKL are $O(cn^3)$, $O(qn^3)$ and $O((q+n)^2n^{2.5})$, respectively, where c is the number of classes, q is the number of kernels.

3) LSSVM l_p MKL: The LSSVM l_p MKL methods are least square SVM-based MKL methods with different normalizations. Similar to the SVM l_p MKL methods, we adopt l_1, l_2 and l_∞ norms to normalize the LSSVM-based MKL methods. The computational complexities of LSSVM l_∞ MKL [38], LSSVM l_1 MKL [53] and LSSVM l_2 MKL [16] are $O(qc^2n^2 + c^3n^3)$, $O(qn^3)$ and $O((q+n)^2(c+n)^{2.5})$, respectively.
4) GP-PMK [54]: The GP-PMK adopts Gaussian Process method and Pyramid Match Kernel to combine multiple kernels to boost the classification performance. The computational complexity of GP-PMK is $O(dn^3)$.
5) LPboost [55]: The LPboost method combines boosting approaches with MKL to mix different kernels. Here we adopt two versions of LPboost, namely LPboost-β and LPboost-B. In LPboost-β, it uses a single vector β to define a combination that works well for all classes jointly. While In LPboost-B, each class can have its own weight vector over the features, in which there is a weight matrix B.
6) ML-KNN [5]: The multi-label k-Nearest Neighbor (ML-KNN) is proposed for single-view multi-label problems. ML-KNN utilizes maximum a posteriori principle to determine the label set for the unseen sample with the help of the statistical information derived from the label sets of an unseen instances neighboring samples. The computational complexity of ML-KNN is $O(nd)$.
7) Simple-MKL [56]: The Simple-MKL is also an SVM-based MKL method, which determines the combinations of different kernels by a reduced gradient algorithm. In our experiments, we apply Simple-MKL for multi-label classification tasks, in which we convert the multi-label problems into several binary classification problems. Its computational complexity is $O(nd^2)$.
8) KLS-CCA [18]: The KLS-CCA formulates the Canonical Correlation Analysis (CCA) as a least-square problem by constructing a specific class indicator matrix. It also converts the multi-label problems into several binary classification problems. Its computational complexity is $O(nc^2 + kc(3n + 5d + 2nd))$, where k is the number of iterations.
9) Hierarchical-SVM: It belongs to the classifier-level method, in which firstly a set of separate SVM are learned for each view and then the results are fused as the input to train another SVM classifier. In our experiment, we also implement Hierarchical-SVM with standard SVM.

B. Multi-Class Classification Tasks

We first evaluate the proposed multi-view algorithm on multi-class classification tasks. The following real word benchmark datasets are used to assess our method as well as the compared methods:

- **NUS-WIDE-OBJECT**: It contains 30,000 real-world object images, falling into 31 classes. In this experiment, we use the official split: 17,927 training images and 12,073 testing images. And we select a set of 26 classes in our experiment as the setting in [57];
- **MSRC-V1**: It contains 240 images with 9 classes. Following the setting in [58], we refine the dataset to get 7 classes and each refined class has 30 images. All the classes include tree, building, airplane, cow, face, car and bicycle.
- **Handwritten Digit**: It contains 0 to 9 ten digit classes and 2,000 data points in total. Five public available features are used in our experiment.

In this subsection, the results of the SVM with each type features and the concatenation of all the features are applied as the baseline. Our method is compared with several competitive SVM-based or LSSVM-based MKL methods (compared method 2, 3) that can make use of multiple types of data. Besides, three more multi-view classification methods are also compared with our method, including GP-PMK [54], LPboost-β and LPboost-B [55], which all perform state-of-the-art performance for multi-class classification tasks.

We conduct 5-fold cross-validation and report the average results with standard deviation. The parameter C in Eq. (2) is optimized in the range of $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4, 10^5\}$. For the SVM and MKL methods, the Gaussian kernel is applied for each type of features (i.e., $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = exp\big(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\big)$), where the parameter γ is finely tuned in the same range as C. We implement the compared MKL methods using the codes published by [16]. Following the setting in [16], in LSSVM l_∞ and l_2 methods, the regularization parameter λ is estimated jointly as the kernel coefficient of an identity matrix; in LSSVM l_1 method, λ is set to 1; in all other SVM approaches, the C parameter of the box constraint is also finely tuned in the same range as C in LMVL. For LPboost-β and LPboost-B methods, we use the codes published by the authors [1]. LIBSVM software package is used to implement all the SVM in our experiments.

C. Experimental Results of Multi-class Classification

In this subsection, as we evaluate the effectiveness of our method for multi-class classification problems, we employ the most widely used criteria, classification accuracy and *Mean Average Precision* (MAP), to evaluate the classification performance. The MAP is the ranking performance computed under each label. The performance of all compared methods in the three classification tasks is reported in Table I.

We can have the following observations regarding performance:

- The results in Table I show that our method generally achieves the best performance against all the compared methods over three datasets in terms of both classification accuracy and MAP (i.e. 0.276, 0.845 and 0.975 evaluated by accuracy; 0.289, 0.864 and 0.989 evaluated by MAP), which validates the effectiveness of our method for multi-view multi-class classification problems.

[1] http://files.is.tue.mpg.de/pgehler/projects/iccv09/

TABLE I

CLASSIFICATION RESULTS OF THE COMPARED METHODS FOR MULTI-CLASS CLASSIFICATION PROBLEMS IN TERMS OF CLASSIFICATION ACCURACY AND MAP (MEAN+STD)

Methods	Auccary (mean+std)			MAP (mean+std)		
	NUS-WIDE	MSRC-V1	Handwritten Digit	NUS-WIDE	MSRC-V1	Handwritten Digit
SVM (Type 1)	0.152±0.018	0.777±0.019	0.943±0.024	0.161±0.016	0.786±0.026	0.964±0.023
SVM (Type 2)	0.149±0.020	0.768±0.018	0.749±0.020	0.152±0.018	0.774±0.022	0.764±0.021
SVM (Type 3)	0.146±0.016	0.781±0.022	0.937±0.019	0.144±0.020	0.794±0.021	0.923±0.018
SVM (Type 4)	0.150±0.018	0.784±0.026	0.935±0.021	0.153±0.019	0.798±0.019	0.958±0.023
SVM (Type 5)	0.141±0.017	0.773±0.023	0.769±0.028	0.142±0.021	0.781±0.018	0.798±0.026
SVM (Type 6)	0.149±0.018	0.789±0.021	-	0.147±0.017	0.799±0.025	-
SVM (All)	0.159±0.020	0.793±0.025	0.948±0.023	0.187±0.021	0.802±0.018	0.969±0.022
SVM l_∞ MKL	0.211±0.023	0.820±0.023	0.954±0.017	0.223±0.019	0.829±0.021	0.975±0.018
SVM l_1 MKL	0.207±0.020	0.813±0.019	0.947±0.024	0.215±0.026	0.824±0.018	0.968±0.023
SVM l_2 MKL	0.202±0.021	0.789±0.022	0.945±0.025	0.212±0.024	0.801±0.022	0.966±0.022
LSSVM l_∞ MKL	0.200±0.018	0.778±0.025	0.948±0.021	0.211±0.020	0.795±0.024	0.969±0.020
LSSVM l_1 MKL	0.195±0.022	0.808±0.027	0.952±0.019	0.198±0.021	0.812±0.026	0.971±0.019
LSSVM l_2 MKL	0.187±0.021	0.796±0.018	0.946±0.024	0.192±0.022	0.819±0.019	0.967±0.022
GP-PMK	0.181±0.020	0.794±0.015	0.942±0.021	0.190±0.016	0.826±0.017	0.969±0.025
LPboost-β	0.220±0.015	0.815±0.010	0.951±0.018	0.229±0.017	0.818±0.022	0.972±0.017
LPboost-B	0.219±0.012	0.813±0.013	0.949 ±0.015	0.227±0.014	0.810±0.022	0.970±0.014
LMVL	**0.276±0.012**	**0.845±0.026**	**0.975±0.021**	**0.289±0.013**	**0.864±0.024**	**0.989±0.019**

- Comparing to the methods that use one single type of features (SVM Type 1 – Type 6), the methods with sophisticated multi-view learning schemes always achieve much better results. For example, MKL-based methods, GP-PMK, LPboost-based methods have better performance over the three datasets, which confirms the usefulness of data integration in multi-class tasks. Comparing to the concatenated method (SVM-All), our method even achieves significant improvements of 11.7%, 5.2% and 2.7% over three datasets in terms of classification accuracy and 10.2%, 6.2% and 2% in terms of MAP, respectively. Moreover, most the multi-view methods compared in our experiment also outperform the concatenated SVM-ALL, which demonstrates that the correlations among views can further boost the performance.
- It is worth noting that our proposed method is still superior to the second best algorithms, which are state-of-the-art multi-view algorithms, over three datasets. For instance, LMVL has a 5.6% performance improvement regarding classification accuracy and 6% performance improvement regarding MAP on NUS-WIDE comparing with the second best results. In MSRC-V1, 2.5% (accuracy) improvement and 3.5% (MAP) improvement are also observed. Margins are reduced on Handwritten Digit dataset (accuracy: 0.975 vs. 0.954 and MAP: 0.989 vs. 0.975).

D. Multi-Label Classification Tasks

In this subsection, we evaluate the proposed multi-view method in multi-view multi-label classification problems, in which each data sample can be associated with more than one label. The effectiveness of proposed method is evaluated on the following datasets:

- **PASCAL VOC 07 (VOC)**: It contains 10,000 images labeled with 20 categories. In this experiment, we use the standard train/test partition, which splits 9,963 images into a training set of 5,011 images and a test set of 4,952 images.
- **MIR Flickr (MIR)**: It contains 25,000 images labeled with 38 categories. In our experiment, images are randomly split into equally sized training and test sets.

The features we used here are from [59]. In this experiment, we choose three representative visual features as different views: local SIFT, global GIST and the tag.

Our method is also compared with several MKL methods used in the previous experiments with the same setting. Besides the MKL methods above, we implement another two popular MKL methods, namely simple-MKL and Hierarchical-SVM. And the trade-off parameter C in simple-MKL and Hierarchical-SVM is finely tuned in the same range in the previous experiments. For our method and MKL methods, we conduct binary classification for each class individually by using one-vs.-others strategy. Instead of SVM, we use the multi-label k-Nearest Neighbor (ML-KNN) [5] to evaluate single-view method for multi-label problems. ML-KNN (Type 1), ML-KNN (Type 2) and ML-KNN (Type 3) denote that the three chosen views, namely SIFT, GIST and tag. These three type of features are separately used to train the ML-KNN for the multi-label prediction. In addition, we implement a kerneled least-square canonical correlation analysis (KLS-CCA) [18]. Mean of the multiple kernels is pre-computed to run this algorithm. The ridge parameter is also finely tuned in the same range used in our method. The different views are fused by combining all the kernels with uniform weights. For the tag features, the linear kernel is used while the Gaussian kernel is adopted for the other two types of features. As the time cost of some compared methods is intolerable, we pre-process the different types of features by PCA to reduce the dimensions.

E. Experimental Results of Multi-label Classification

In our method, as we classify each class independently using the one-vs.-others strategy, two popular evaluation criteria for multi-label classification, namely *Mean Average Precision*

TABLE II

CLASSIFICATION RESULTS OF THE COMPARED METHODS FOR MULTI-LABEL CLASSIFICATION PROBLEMS IN TERMS OF MAP AND HL (MEAN+STD)

Methods	Map (mean+std)		HL (mean+std)	
	PASCAL VOC 07	MIR Flickr	PASCAL VOC 07	MIR Flickr
ML-KNN (Type 1)	0.511±0.002	0.457±0.003	0.069±0.002	0.120±0.001
ML-KNN (Type 2)	0.514±0.001	0.461±0.005	0.067±0.002	0.122±0.002
ML-KNN (Type 3)	0.513±0.002	0.459±0.005	0.064±0.001	0.118±0.001
ML-KNN (ALL)	0.521±0.002	0.471±0.003	0.065±0.001	0.116±0.001
Simple-MKL	0.587±0.007	0.523±0.004	0.061±0.001	0.112±0.003
SVM l_∞ MKL	0.598±0.005	0.530±0.007	0.056±0.002	0.103±0.001
SVM l_1 MKL	0.594±0.006	0.527±0.005	0.057±0.001	0.105±0.002
SVM l_2 MKL	0.588±0.005	0.517±0.008	0.059±0.001	0.106±0.001
LSSVM l_∞ MKL	0.583±0.006	0.524±0.007	0.058±0.002	0.107±0.001
LSSVM l_1 MKL	0.579±0.006	0.516±0.009	0.059±0.001	0.109±0.002
LSSVM l_2 MKL	0.574±0.005	0.526±0.010	0.061±0.002	0.110±0.001
KLS-CCA	0.537±0.009	0.509±0.008	0.059±0.001	0.117±0.001
Hierarchical-SVM	0.562±0.015	0.504±0.011	0.060±0.004	0.109±0.005
LMVL	**0.608±0.008**	**0.536±0.007**	**0.053±0.000**	**0.094±0.001**

(MAP) and *Hamming Loss* (HL), are adopted rather than traditional classification accuracy criterion. HL is used to evaluate the label set predictions for each instance. It is the fraction of labels that are incorrectly predicted. Note that a smaller value in HL indicates better performance. The performance of the compared methods on the VOC dataset and MIR dataset are reported in Table II. Both the mean and standard deviation of two criteria are presented.

From the experimental results, we observe that :

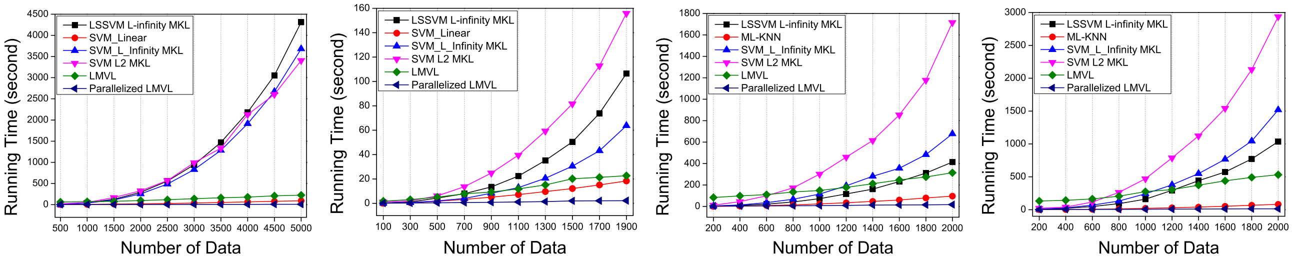
- As shown in table II, our method still performs the best for the multi-view multi-label problem against all the compared methods over the two datasets, which further validates the effectiveness of our method for multi-label classification problems.
- The methods using multiple types of features generally perform better than the methods using a single type of features, which also verifies the point again that the consistent and complementary information indeed can improve the models' performance. Although ML-KNN method is designed for the multi-label problems, it can only work with one type of features. Concatenating all the features as input, ML-KNN treats each type of features homogeneously without distinctions. All the multi-view methods in the feature-level fusion or classifier-level fusion always achieve better performance, which indicates that appropriately fusing features from different views is significant for improving the performance of tasks.
- Compared with the classifier-level method, Hierarchical-SVM, the other multi-view methods in feature-level fusion can achieve better performance, which is consistent with the point mentioned above that feature-level fusion tends to be more effective than the classifier-level fusion as richer information is contained.
- In comparison with KLS-CCA, our method performs better as the raw information from different views is well preserved. For example, our method achieves improvements of 7.1% and 2.3% over the two datasets in terms of MAP.
- Compared with the improvement in previous multi-class classification tasks, the improvements in this subsection are reduced (MAP: 0.608 vs. 0.598 and 0.536 vs. 0.530; HL: 0.053 vs. 0.056 and 0.094 vs. 0.103).

F. Convergence Tests

To evaluate the efficiency of LMVL, we conduct an experiment in which the running time results of some representative algorithms with respect to different numbers of data samples are reported in Figure 1. Here, we choose linear SVM with all the features, ML-KNN with all the features, SVM l_2 MKL, SVM l_∞ MKL, LSSVM l_∞ MKL, LMVL and Parallelized LMVL. As shown in Figure 1, the computational complexities of SVM-based and MKL-based methods with different norms are non-linear while our method just has a linear computational cost. The difference of running time between non-linear methods and our proposed method is significant when learning large-scale multi-view datasets. Furthermore, it is worth noting that the Parallelized LMVL's computational cost is lower than SVM-Linear and ML-KNN, which demonstrates that our method is applicable to large-scale multi-view datasets.

CONCLUSIONS

In this paper, we propose a linear multi-view learning framework for multi-class and multi-label classification problems with convergence guarantee. To learn all the features from multiple views, we develop an SVM-based multi-view learning framework. In our model, it can automatically select more important views for subsequent classifier to predict each class. Since the objective function is non-smooth and difficult to solve, we propose an efficient optimization method by converting a multi-view learning problem to a set of linear single-view learning problems. As a consequence, instead of directly learning the entire weight matrix for different views, the proposed algorithm can separately learn each column of the projection matrix for each class, which can be easily parallelized on a multi-core machine. Besides, our method is applicable to large-scale multi-view learning problems as it has a linear computational cost. Furthermore, theoretical convergence proof of the algorithm is also provided. Extensive experiments have been conducted on several benchmark datasets. Experimental results show that our algorithm is capable of both multi-class and multi-label classification tasks on large-scale datasets.



(a) NUS-OBJECT-WIDE (b) Handwritten Digit (c) PASCAL VOC 07 (d) MIR

Fig. 1. Training time of different methods with respect to different number of samples obtained from four datasets. From this figure, we can see that LMVL is an efficient algorithm with a linear computational cost.

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1. IEEE, 2005, pp. 886–893.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *ECCV 2006*, pp. 404–417, 2006.

[3] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, vol. 2. Ieee, 1999, pp. 1150–1157.

[4] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Machine learning*, vol. 73, no. 2, pp. 133–153, 2008.

[5] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *PR*, vol. 40, no. 7, pp. 2038–2048, 2007.

[6] S. Wang, X. Chang, X. Li, Q. Z. Sheng, and W. Chen, "Multi-task support vector machines for feature selection with shared knowledge discovery," *Signal Processing*, vol. 120, pp. 746–753, 2016.

[7] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, "Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding," *PR*, vol. 48, no. 10, pp. 3102–3112, 2015.

[8] B. Du, Z. Wang, L. Zhang, L. Zhang, W. Liu, J. Shen, and D. Tao, "Exploring representativeness and informativeness for active learning," *IEEE transactions on cybernetics*, vol. 47, no. 1, pp. 14–26, 2017.

[9] X. Xue, F. Nie, S. Wang, X. Chang, B. Stantic, and M. Yao, "Multi-view correlated feature learning by uncovering shared component," in *Thirty-First AAAI*, 2017.

[10] H. Q. Minh, L. Bazzani, and V. Murino, "A unifying framework in vector-valued reproducing kernel hilbert spaces for manifold regularization and co-regularized multi-view learning," *JMLR*, vol. 17, no. 25, pp. 1–72, 2016.

[11] L. Zhu, J. Shen, X. Liu, L. Xie, and L. Nie, "Learning compact visual representation with canonical views for robust mobile landmark search." IJCAI, 2016.

[12] Y. Luo, D. Tao, C. Xu, C. Xu, H. Liu, and Y. Wen, "Multiview vector-valued manifold regularization for multilabel image classification," *IEEE TNNLS*, vol. 24, no. 5, pp. 709–722, 2013.

[13] S. Ji, L. Sun, R. Jin, and J. Ye, "Multi-label multiple kernel learning," in *NIPS*, 2009, pp. 777–784.

[14] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *JMLR*, vol. 12, no. Jul, pp. 2211–2268, 2011.

[15] M. Kloft, U. Brefeld, P. Laskov, and S. Sonnenburg, "Non-sparse multiple kernel learning," in *NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, vol. 4, 2008.

[16] S. Yu, T. Falck, A. Daemen, L.-C. Tranchevent, J. A. Suykens, B. De Moor, and Y. Moreau, "L 2-norm multiple kernel learning and its application to biomedical data fusion," *BMC Bioinformatics*, vol. 11, no. 1, p. 1, 2010.

[17] A. Shon, K. Grochow, A. Hertzmann, and R. P. Rao, "Learning shared latent structure for image synthesis and robotic imitation," in *NIPS*, 2005, pp. 1233–1240.

[18] L. Sun, S. Ji, and J. Ye, "Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis," *IEEE TPAMI*, vol. 33, no. 1, pp. 194–200, 2011.

[19] Q. Chen and S. Sun, "Hierarchical multi-view fisher discriminant analysis," in *ICONIP*. Springer, 2009, pp. 289–298.

[20] C. Hou, C. Zhang, Y. Wu, and F. Nie, "Multiple view semi-supervised dimensionality reduction," *PR*, vol. 43, no. 3, pp. 720–730, 2010.

[21] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint 2, 1-norms minimization," in *NIPS*, 2010, pp. 1813–1821.

[22] H. Wang, F. Nie, and H. Huang, "Multi-view clustering and feature learning via structured sparsity." in *ICML*, 2013, pp. 352–360.

[23] T. Liu, D. Tao, M. Song, and S. Maybank, "Algorithm-dependent generalization bounds for multi-task learning." *IEEE TPAMI*, 2016.

[24] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[25] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE TPAMI*, vol. 38, no. 3, pp. 447–461, 2016.

[26] F. Nie, J. Li, X. Li *et al.*, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification." IJCAI, 2016.

[27] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *PR*, vol. 37, no. 9, pp. 1757–1771, 2004.

[28] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *NIPS*, 2001, pp. 681–687.

[29] A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data," in *PKDD*. Springer, 2001, pp. 42–53.

[30] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbours." in *AAAI*, 2017, pp. 2408–2414.

[31] S. Wang, Y. Yang, Z. Ma, X. Li, C. Pang, and A. G. Hauptmann, "Action recognition by exploring data distribution and feature correlation," in *CVPR, 2012 IEEE Conference on*. IEEE, 2012, pp. 1370–1377.

[32] L. Zhu, J. Shen, H. Jin, R. Zheng, and L. Xie, "Content-based visual landmark search via multimodal hypergraph learning," *IEEE transactions on cybernetics*, vol. 45, no. 12, pp. 2756–2769, 2015.

[33] C. Xu, D. Tao, and C. Xu, "Multi-view self-paced learning for clustering," in *AAAI*, 2015, pp. 3974–3980.

[34] Z. Zhang, Z. Zhai, and L. Li, "Uniform projection for multi-view learning," *IEEE TPAMI*, 2016.

[35] Y. Luo, T. Liu, D. Tao, and C. Xu, "Multiview matrix completion for multilabel image classification," *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2355–2368, 2015.

[36] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *JMLR*, vol. 5, no. Jan, pp. 27–72, 2004.

[37] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *JMLR*, vol. 7, no. Jul, pp. 1531–1565, 2006.

[38] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "Lp-norm multiple kernel learning," *JMLR*, vol. 12, no. Mar, pp. 953–997, 2011.

[39] M. White, X. Zhang, D. Schuurmans, and Y.-l. Yu, "Convex multi-view subspace learning," in *NIPS*, 2012, pp. 1673–1681.

[40] R. Memisevic, L. Sigal, and D. J. Fleet, "Shared kernel information embedding for discriminative inference," *IEEE TPAMI*, vol. 34, no. 4, pp. 778–790, 2012.

[41] L. Zhu, J. Shen, H. Jin, L. Xie, and R. Zheng, "Landmark classification with hierarchical multi-modal exemplar feature," *IEEE TMM*, vol. 17, no. 7, pp. 981–993, 2015.

[42] G. Fumera and F. Roli, "A theoretical and experimental analysis of linear combiners for multiple classifier systems," *IEEE TPAMI*, vol. 27, no. 6, pp. 942–956, 2005.

[43] C. G. Snoek, M. Worring, and A. W. Smeulders, "Early versus late fusion in semantic video analysis," in *ACM MM*. ACM, 2005, pp. 399–402.

[44] S. Wang, Z. Ma, Y. Yang, X. Li, C. Pang, and A. G. Hauptmann, "Semi-supervised multiple feature analysis for action recognition," *IEEE TMM*, vol. 16, no. 2, pp. 289–298, 2014.

[45] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*. ACM, 1998, pp. 92–100.

[46] B. S. Parker and L. Khan, "Detecting and tracking concept class drift and emergence in non-stationary fast data streams." in *AAAI*, 2015, pp. 2908–2913.

[47] U. Brefeld and T. Scheffer, "Co-em support vector learning," in *ICML*. ACM, 2004, p. 16.

[48] I. Muslea, S. Minton, and C. A. Knoblock, "Active+ semi-supervised learning= robust multi-view learning," in *ICML*, vol. 2, 2002, pp. 435–442.

[49] A. P. James and B. V. Dasarathy, "Medical image fusion: A survey of the state of the art," *Information Fusion*, vol. 19, pp. 4–19, 2014.

[50] X. Cai, F. Nie, H. Huang, and C. Ding, "Multi-class l2, 1-norm support vector machine," in *IEEE ICDM*. IEEE, 2011, pp. 91–100.

[51] F. Nie, Y. Huang, X. Wang, and H. Huang, "New primal svm solver with linear computational cost for big data classifications," in *ICML*, 2013.

[52] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[53] J. A. Suykens, T. Van Gestel, and J. De Brabanter, *Least Squares Support Vector Machines*. World Scientific, 2002.

[54] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Gaussian processes for object categorization," *IJCV*, vol. 88, no. 2, pp. 169–188, 2010.

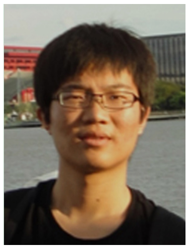
[55] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *IEEE ICCV*. IEEE, 2009.

[56] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *JMLR*, vol. 9, no. Nov, pp. 2491–2521, 2008.

[57] S. Gao, L.-T. Chia, and I. W.-H. Tsang, "Multi-layer group sparse codingfor concurrent image classification and annotation," in *CVPR*. IEEE, 2011, pp. 2809–2816.

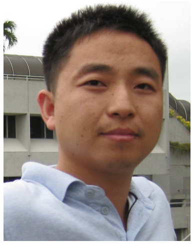
[58] H. Wang, F. Nie, H. Huang, and C. Ding, "Heterogeneous visual features fusion via sparse multimodal machine," in *IEEE CVPR*, 2013, pp. 3097–3102.

[59] M. Guillaumin, J. Verbeek, and C. Schmid, "Multimodal semi-supervised learning for image classification," in *IEEE CVPR*. IEEE Computer Society, 2010, pp. 902–909.

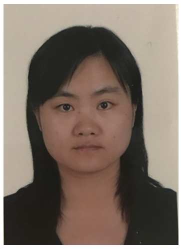


Xiaowei Xue received his B.Sc. degree from Northeastern University in 2011. He is currently pursuing Ph.D. degree at the college of Computer Science and Technology, Zhejiang University, Hangzhou, China.

His research fields are Multi-view learning, computational intelligence.



Feiping Nie received the Ph.D. degree in computer science from Tsinghua University, Beijing, China in 2009. He is currently a Professor with Center for OPTical Imagery Analysis and Learning, Northwestern Polytechnical University, Shaanxi, China. His research interests are machine learning and its applications fields, such as pattern recognition, data mining,computer vision, image processing and information retrieval. He has published more than 100 papers in the prestigious journals and conferences like TPAMI, TKDE, ICML, NIPS, KDD, and etc. He is now serving as Associate Editor or PC member for several prestigious journals and conferences in the related fields.



Zhihui Li received the B.S. degree from Beijing University of Posts and Telecommunications in 2008. She is currently working as a Data Analyst in Beijing Etrol Technologies Co., Ltd.

Her research interests include artificial intelligence, machine learning, and computer vision.

Sen Wang received his PhD from the University of Queensland (UQ), Australia in 2014. During Jan 2014 and Jul 2016, he was an ARC postdoctoral research fellow in DKE group in UQ. Currently, he is a lecturer in School of Information and Communication Technology in Griffith University. Sen Wangs major areas of research interests include: medical data analysis, signal and image processing, pattern recognition and machine learning algorithms, big data analytics.

Xue Li is a full professor in the School of Information Technology and Electrical Engineering at the University of Queensland, Australia. He is an Adjunct Professor at Chongqing University, China. He obtained his PhD degree in Information Systems from Queensland University of Technology in 1997. Dr Xue Lis research areas are Big Data Analytics, Pattern Recognition, and Intelligent Information Systems. He is a member of ACM and IEEE.

Min Yao received his Ph.D. degree in Biomedical Engineering and Instrument from Zhejiang University, China, in 1995. He is currently a professor at the college of Computer Science and Technology, Zhejiang University. His research interests include computational intel- ligence, pattern recognition, knowledge discovery and knowledge service.