Adaptive-weighting discriminative regression for multi-view classification

Muli Yang, Cheng Deng, Feiping Nie

School of Electronic Engineering, Xidian University, Xi’an 710071, China
State Key Laboratory of Integrated Services Networks, Xidian University, China
Center for OPTical Imagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi’an 710072, China

ARTICLE INFO

Article history:
Received 27 July 2018
Revised 14 October 2018
Accepted 17 November 2018
Available online 22 November 2018

Keywords:
Multi-view learning
Supervised learning
Classification

ABSTRACT

Multi-view data represented by different features have been involved in many machine learning applications. Efficiently exploiting and preserving the correlative yet complementary information in multiple views remains challenging in multi-view learning. Comparing with existing methods that separately cope with each view, we propose a supervised multi-view feature learning framework to handle diverse views with a unified perception. Specifically, we fuse the multi-view data by mapping the concatenation of original features to a discriminative low-dimensional subspace, where the features from different views are adaptively assigned with the learned optimal weights. This strategy can simultaneously preserve the correlative and the complementary information, which is further enhanced to be more discriminative for subsequent classification. An efficient iterative algorithm is devised to optimize the formulated framework with closed-form solutions. Comprehensive evaluations with several state-of-the-art competitors demonstrate the efficiency and the superiority of the proposed method.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Due to the development of information technology, multi-view data have met a widespread increase over the last few decades. For instance, an image can be represented by a variety of features, e.g., GIST [1], LBP [2], SIFT [3], HOG [4], and Color [5,6]. Since these features naturally reflect different descriptions of the image from different views, the goal of multi-view learning is to exploit the correlative yet complementary information of these different views. Efficiently fusing the information from diverse views has always been a challenging issue in multi-view learning. During this decade, much progress has been made in addressing this issue with unsupervised and semi-supervised manners. Some methods conduct multi-view learning based on classical co-training [7–10] and co-regularization [11–14] approaches, which carry out satisfactory fusions but ignore the problems induced by high-dimensional multi-view data, e.g., over-fitting and expensive computational cost. Considering the impact of high-dimensional multi-view data, some other methods [15–17] perform multi-view learning based on Canonical Correlation Analysis (CCA), which maximizes the correlations across different views in a latent subspace.

Despite the fact that these methods can effectively project high-dimensional features into a low-dimensional subspace, they merely focus on the correlations across different views while neglect the complementary information of each view, which may result in inefficient fusions of the original multi-view data.

Benefiting from the label information, supervised multi-view learning has received more and more attention in recent years. Practically, Arora and Livescu [18] combined a series of CCA and Linear Discriminant Analysis (LDA) models to learn the bottleneck features when the training set is phonetically labeled. Liu et al. [19] proposed a multi-view logistic regression framework, taking advantage of multiple Hessian regularization obtained from each particular view. Another method [20] learns multiple view-specific projections guided by the label information, which maps the view-wise features into a group of low-dimensional subspaces. Essentially, these methods follow the intention of Multiple Kernel Learning (MKL) methods [21] to conduct “decision-level” fusions (fusing after view-wise processing), which preserve the complementary information of each view but ignore the correlations across diverse views.

In this paper, we propose a novel supervised multi-view feature learning framework, which is able to preserve both the correlative and the complementary information of the original views. Specifically, we formulate our framework with a regression-based structure and employ a new discriminative regression target to preserve
and enhance the discrimination of the features in the projected subspace. Based on the fact that different views often contribute unequally in various tasks, a set of learnable weight parameters are merged into the transformation matrix to conduct efficient fusion of diverse views. With the aid of the adaptive weights, the correlative information and the complementary one can be simultaneously preserved in the projected discriminative subspace. In contrast to the work [20] that involves multiple projections, our method learns one adaptive-weighting discriminative projection, which simplifies the learning process as well as guarantees a unified perception. An efficient iterative algorithm is devised to optimize the formulated framework jointly, which ensures robust discrimination when learning the correlative and the complementary features. We evaluate our method on ten broadly used benchmark datasets, with each composed of up to six types of popular visual features. Comprehensive experimental results of the proposed method and several state-of-the-art competitors can demonstrate the superiority of our proposed method.

To sum up, the contributions of this paper are three-fold:

(1) We address the multi-view feature learning problem with a novel discriminative regression based framework, which maps the multi-view data to a unified low-dimensional discriminative subspace.

(2) We introduce a set of learnable weight parameters which can be merged into the transformation matrix, such that the correlative and the complementary information of the original views can be preserved in the projected subspace simultaneously.

(3) We design an efficient iterative optimization algorithm with closed-form solution to update the learnable parameters during each iteration, which expresses a remarkable convergence speed in extensive experiments.

The remainder of this paper is organized as follows. We provide the notations and background in Section 2. In Section 3, we present the proposed multi-view learning framework, followed by an iterative algorithm exploited to optimize the proposed method in Section 4. Section 5 exhibits the experimental results, and conclusions are drawn in Section 6.

2. Background

In this section, we will introduce the notations throughout the paper, and then briefly review several representative multi-view learning methods.

2.1. Notations

In this paper, vectors and matrices are written in boldface, and respectively with lowercase and uppercase letters. To denote the jth element of a vector m, we use the notation of m(j). When it comes to a matrix M, the form of Mij is adopted to denote the ith row and jth column element. Furthermore, the ℓ2-norm of a vector is denoted as ∥·∥2, and the Frobenius norm of a matrix is denoted as ∥·∥F. For brevity, the all-one column vector (or matrix) is denoted as 1, whose size is specified by the subscript. Similarly, the zero vector (or matrix) is denoted as 0, with an appropriate number of zero elements. The other frequently-used notations throughout the paper are summarized in Table 1.

Table 1
Notations.  
<table>
<thead>
<tr>
<th>Notations</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>Number of data samples</td>
</tr>
<tr>
<td>c</td>
<td>Number of classes</td>
</tr>
<tr>
<td>v</td>
<td>Number of views</td>
</tr>
<tr>
<td>d</td>
<td>Dimensionality of the kth view</td>
</tr>
<tr>
<td>y</td>
<td>Label of the corresponding sample x_i</td>
</tr>
<tr>
<td>w</td>
<td>Weight of the kth view</td>
</tr>
<tr>
<td>X^{(k)}_i</td>
<td>Feature vector of the i-th sample in the k-th view</td>
</tr>
<tr>
<td>y</td>
<td>Coded label vector of the i-th sample</td>
</tr>
<tr>
<td>m</td>
<td>Adjustment vector of the i-th sample</td>
</tr>
<tr>
<td>X</td>
<td>Feature matrix of all samples in the k-th view</td>
</tr>
<tr>
<td>x</td>
<td>Feature matrix of all samples across all v views</td>
</tr>
<tr>
<td>y</td>
<td>Coded label matrix of all samples</td>
</tr>
<tr>
<td>m</td>
<td>Adjustment matrix of all samples</td>
</tr>
<tr>
<td>w</td>
<td>Transformation matrix of all views</td>
</tr>
<tr>
<td>W</td>
<td>Weighted transformation matrix across all views</td>
</tr>
</tbody>
</table>

2.2. CCA

Canonical Correlation Analysis (CCA) [22–24] is a well-known multi-view learning algorithm that aims to obtain a hypothetical latent subspace shared by multiple views, from which these different views can be generated. Basically, CCA seeks linear transformations for features in each view, such that the correlation between the transformed feature sets is maximized in a common latent subspace, while the self covariance of each transformed feature set is regularized to be small enough.

Practically, for X_1 ∈ R^d_1×n and X_2 ∈ R^d_2×n in a two-view scenario, two transformation vectors, w_1 ∈ R^d_1 and w_2 ∈ R^d_2, are computed to maximize the following correlation coefficient ρ:

$$\rho = \frac{w_1^T X_1 X_2^T w_2}{\sqrt{(w_1^T X_1 X_1^T w_1)(w_2^T X_2 X_2^T w_2)}}.$$  (1)

Since w_1 and w_2 are scale-independent towards ρ, CCA can be equivalently depicted as the following optimization problem:

$$\max_{w_1, w_2} w_1^T X_1 X_2 w_2$$

s.t.  
$$w_1^T X_1 X_1^T w_1 = w_2^T X_2 X_2^T w_2 = 1.$$  (2)

The optimal solution to the transformation vectors w_1 and w_2 can be derived through solving a generalized eigenvalue problem:

$$\begin{bmatrix} 0 & X_1 X_1^T \\ X_2 X_2^T \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \lambda \begin{bmatrix} 1 & X_1 X_1^T \\ 0 & X_2 X_2^T \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}.$$  (3)

CCA is originally implemented in the unsupervised scenario. With the introduction of a set of regularization hyper-parameters, Sharma et al. [25] proposed Generalized Multi-view Analysis (GMA) to extend CCA in a supervised setting. In contrast, Kan et al. [26] proposed Multi-view Discriminant Analysis (MvDA) without any hyper-parameter, which can simultaneously obtain multiple transformation matrices from different views by solving a single eigenvalue problem.

2.3. MKL

Since kernels in Multiple Kernel Learning (MKL) can naturally correspond to different views, MKL has been applied with great success to cope with the multi-view data by combining kernels appropriately.

SimpleMKL [21,27] is a representative and efficient MKL based algorithm. Apart from learning the combination of kernels, SimpleMKL solves a standard SVM optimization problem, where the kernel is defined as a linear combination of multiple kernels. Specifically, SimpleMKL aims to solve the following primal problem:

$$\min_{w_i, b, \xi} \sum_{k=1}^{v} \frac{1}{\alpha_k} \|w_k\|^2 + C \sum_{i=1}^{n} \xi_i$$

s.t.  
$$y_i \left( \sum_{k=1}^{v} w_k x_i^{(k)} + b \right) \geq 1 - \xi_i.$$
\[
\sum_{k=1}^{n} a_k = 1, \xi_i \geq 0, a_k \geq 0 (\forall i, k),
\]

where \(a_k\) controls the smoothness of each corresponding kernel function, and the \(\ell_1\)-norm constraint on \(a\) is a sparsity regularization to encourage fewer basis kernels by forcing some \(a_k\) of \(\{a_k\}_{k=1}^n\) to be zero. Therefore, the optimal SVM objective function value \(f(a)\) can be defined as

\[
J(a) = \left\{ \min_{w, b, \xi} \sum_{i=1}^{n} \frac{1}{2} \|w_i\|^2 + C \sum_{i=1}^{n} \xi_i \right. \\
\left. \text{s.t. } y_i \left( \sum_{k=1}^{n} w_k x_i^{(k)} + b \right) \geq 1 - \xi_i, \xi_i \geq 0 (\forall i) \right. \}
\]

The primal optimization problem is then reformulated as

\[
\min_a J(a) \text{ s.t. } \sum_{k=1}^{n} a_k = 1, a_k \geq 0 (\forall k),
\]

which can be solved with a two-step method: (1) solving a conventional SVM optimization problem \(J(a)\) with given \(a\); (2) updating \(a\) by gradient descent when the \(\ell_1\)-norm constraint on \(a\) is satisfied.

3. The proposed method

Given an \(n\)-sample dataset \(\{x_i, y_i\}_{i=1}^{n}\), where each sample \(x_i \in \mathbb{R}^d\) has \(d\)-dimensional features, \(y_i \in \{1, 2, \ldots, c\}\) is the corresponding label information of \(x_i\), with \(c \geq 2\) classes. The common used least squares regression for single view classification can be addressed as the following optimization problem:

\[
\min_{\mathbf{W}_1} \sum_{i=1}^{n} \| \mathbf{W}_1^T x_i - \mathbf{b} - y_i \|^2 + \lambda \| \mathbf{W}_1 \|^2,
\]

where \(\mathbf{W} \in \mathbb{R}^{d \times c}\) is a transformation matrix, \(\mathbf{b} \in \mathbb{R}^c\) is an intercept vector, and \(\lambda\) is a trade-off parameter. In multi-class scenario, \(y_i \in \mathbb{R}^c\) is often coded as \([-1, -1, +1, -1, -1, -1, \ldots]^{T}\) with only the \(y_i\)th element being “+1” and others being “-1” to identify the \(c\) different classes uniquely.

As for the multi-view setting, we denote the sample \(x_i\) in the \(k\)th view (\(v\) views in total) as \(x_i^{(k)} \in \mathbb{R}^{d_k}\), where \(d_k\) is the feature dimension of the \(k\)th view. The corresponding transformation matrix is thus denoted as \(\mathbf{W}_k \in \mathbb{R}^{d_k \times c}\). Considering that the elements of the regression target \(\mathbf{y}_i\) in Eq. (7) are fixed as +1 or -1, which usually causes wrong penalization to the right classifications that are far from ±1, we present our multi-view learning framework with a new discriminative regression target and a set of weight parameters to fit the multi-view classification scenario as follow

\[
\min_{\mathbf{W}, \mathbf{b}, \alpha} \sum_{i=1}^{n} \sum_{k=1}^{v} \sqrt{\alpha_k} \| \mathbf{W}_k^T x_i^{(k)} + b - y_i \|^2 + \lambda \sum_{k=1}^{v} \| \mathbf{W}_k \|^2,
\]

where \(\{\alpha_k\}_{k=1}^{v}\) is a set of positive weights with the restriction of \(\sum_{k=1}^{v} \alpha_k = 1\). The new discriminative regression target, i.e., \(\mathbf{r} = \mathbf{y}_i + \mathbf{y}_i \odot \mathbf{m}\), is formulated as a simple yet more efficient variant than the one in Xiang et al. [28] where another parameter is involved. Here the symbol “\(\odot\)” indicates the Hadamard product operation, and \(\mathbf{m}\) is a non-negative adjustment vector. This new regression target is a natural extension of the original regression target \(\mathbf{y}_i\) in Eq. (7). Since the value of \(\mathbf{y}_i\) is either +1 or -1, the non-negativity of the adjustment vector \(\mathbf{m}\) can always increase the absolute value of the new regression target \((\mathbf{y}_i + \mathbf{y}_i \odot \mathbf{m})\). Given a set of samples \(\{x_i, y_i\}\) falling into 3 different classes, for instance, the discriminative regression target of \(x_i\) belonging to the second class can be depicted as

\[
\tilde{y}_i^{(1)} \leftrightarrow (-1 - m_i^{(1)}) \quad \tilde{y}_i^{(2)} \leftrightarrow (+1 + m_i^{(2)}) \quad \tilde{y}_i^{(3)} \leftrightarrow (-1 - m_i^{(3)})
\]

where \([\tilde{y}_i^{(1)}, \tilde{y}_i^{(2)}, \tilde{y}_i^{(3)}]^{T} = \hat{y}_i = \mathbf{W} \mathbf{x}_i + \mathbf{b}\) is the predicted label vector for the corresponding \(x_i\), and \([m_i^{(1)}, m_i^{(2)}, m_i^{(3)}]^T = \mathbf{m}\), is the adjustment vector with each element being non-negative. This strategy can preserve and enlarge the distances between different classes in the projected subspace, which ensures a more robust performance than the conventional least squares regression.

For brevity, we collect all samples in the \(k\)th view into \(X_k = [x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(n_k)}] \in \mathbb{R}^{d_k \times n_k}\). Similarly, \(Y = [y_1, y_2, \ldots, y_n]^{T} \in \mathbb{R}^{n \times c}\) records the corresponding labels, and \(\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_n] \in \mathbb{R}^{n \times c}\) contains the associated adjustment vectors. Then the optimization problem in Eq. (8) can be reformulated as

\[
\min_{\mathbf{W}, \mathbf{b}, \mathbf{M}, \alpha} \sum_{k=1}^{n} \sqrt{\alpha_k} \| \mathbf{W}_k^{(1)} x_k^{(1)} + 1_a \mathbf{b}^{(1)} - \mathbf{Y} - \mathbf{Y} \odot \mathbf{M} \|_F^2 + \lambda \sum_{k=1}^{n} \| \mathbf{W}_k \|_2^2,
\]

where \(1_a = [1, 1, 1, \ldots]^{T} \in \mathbb{R}^n\) is an all-one vector.

However, Eq. (9) is still hard to be optimized. A further step is required to ensure the efficiency of the optimization procedure with the introduction of parameter \(\alpha\). As the two items of the objective function in Eq. (9) can be alternatively optimized, we merge the parameter \(\alpha\) into the transformation matrix as \(\sqrt{\alpha_k} \mathbf{W}_k = \mathbf{W}_k^{(1)}\). Therefore, the second item of Eq. (9) can be rewritten by introducing Cauchy inequality:

\[
\lambda \sum_{k=1}^{n} \| \mathbf{W}_k \|_2^2 = \lambda \sum_{k=1}^{n} \frac{1}{\alpha_k} \| \mathbf{W}_k \|_2^2 \geq \lambda \left( \sum_{k=1}^{n} \| \mathbf{W}_k \|_2 \right)^2.
\]

The second equation of Eq. (10) is satisfied, if and only if, the weight parameter \(\alpha_k\) takes a certain value, which brings an optimal closed-form solution when solving the minimization problem in the optimization procedure described in Section 4. It is noteworthy that we employ the square root form of parameter \(\alpha_k\) in Eqs. (8) and (9) to ensure the feasibility of the above further step. Accordingly, our proposed multi-view learning framework can be derived from the above analysis and depicted as

\[
\min_{\mathbf{W}, \mathbf{b}, \mathbf{M}, \alpha} \left\| \sum_{k=1}^{n} \mathbf{W}_k^{(1)} \mathbf{x}_k^{(1)} + 1_a \mathbf{b}^{(1)} - \mathbf{Y} - \mathbf{Y} \odot \mathbf{M} \right\|_F^2 + \lambda \sum_{k=1}^{n} \| \mathbf{W}_k \|_2^2 \text{ s.t. } \mathbf{M} > 0, \sum_{k=1}^{n} \alpha_k = 1, \forall \alpha_k > 0.
\]

We can see at least four advantages of the proposed multi-view learning framework presented in Eq. (11). First, a new discriminative regression target is introduced to address the multi-view learning tasks, which helps to learn intensive robust discriminative features in the projected subspace. Second, with the adaptive weight parameter \(\alpha\), the contributions of different views can be arranged adaptively for a better representation and fusion of the original features. Third, the weight parameter \(\alpha\) is merged into the transformation matrix, with which a unified projection with adaptive weights can be learned directly. Fourth, the formulation of the proposed framework can be optimized efficiently with optimal closed-form solutions, which will be explored in the following Section 4.

4. Optimization

According to relevant convex optimization theory, it is easy to be justified that both items in Eq. (11) are convex (the Hessian...
matrix of the second item can prove to be semi-positive), indicating problem (11) is thereby convex. The convexity implies the existence of the optimal solutions, which are presented in the following three theorems. An iterative algorithm is thus exploited to guide the training process using the four derived optimal solutions during each iteration, with rigorous theoretical proof on the convergence.

4.1. Optimize \( \tilde{W} \) and \( b \)

For simplicity, we collect \( X_0 \) and \( \tilde{W}_0 \) of all \( v \) views into \( X = [X_1, X_2, \ldots, X_d] \in \mathbb{R}^{d \times n} \) and \( \tilde{W} = [\tilde{W}_1, \tilde{W}_2, \ldots, \tilde{W}_n] \in \mathbb{R}^{d \times c} \), where \( d = \sum_{v=1}^{V} d_v \) is the dimension of features across all \( v \) views. Fixing \( \alpha \) and \( M \), the optimization of problem (11) with respect to \( \tilde{W} \) and \( b \) can be derived from the following theorem.

**Theorem 1.** Given fixed parameters \( \alpha \) and \( M \), the optimal solutions of problem (11) with respect to \( \tilde{W} \) and \( b \) can be calculated as

\[
\tilde{W} = (XNX^T + \lambda A)^{-1}XN(Y + Y \odot M),
\]

\[
b = \frac{1}{n}(Y + Y \odot M)1_n - \frac{1}{n} \tilde{W}^T X 1_n,
\]

where \( A = \text{diag}(\frac{1}{\alpha_1}, \ldots, \frac{1}{\alpha_1}, \frac{1}{\alpha_2}, \ldots, \frac{1}{\alpha_2}, \ldots, \frac{1}{\alpha_v}, \ldots, \frac{1}{\alpha_v}) \in \mathbb{R}^{d \times d} \) is a diagonal matrix in which the \( i \)th diagonal element records the reciprocal of the weight \( \alpha \) corresponding to the \( i \)th feature in \( X \), and \( N = I_n - \frac{1}{n} 1_n 1_n^T \) is an identity matrix and \( 1_n \) is an \( n \times 1 \) all-one matrix.

**Proof.** Since parameters \( \alpha \) and \( M \) are fixed, as stated in Eq. (11), the optimal \( \tilde{W} \) and \( b \) can be derived by minimizing the objective function

\[
f(\tilde{W}, b) = \|X^T \tilde{W} + b 1_n 1_n^T - Y \odot M\|^2_F + \frac{\lambda}{\alpha} \|\tilde{W}\|^2_F.
\]

We solve this problem by setting the partial derivations of \( f(\tilde{W}, b) \) to 0, i.e.,

\[
\frac{\partial f(\tilde{W}, b)}{\partial b} = 0, \quad \frac{\partial f(\tilde{W}, b)}{\partial \tilde{W}} = 0.
\]

From the first equation, we get

\[
\tilde{W}^T X 1_n + b 1_n 1_n^T - (Y + Y \odot M)1_n = 0.
\]

Thus we can derive the \( b = \frac{(Y + Y \odot M)1_n - \tilde{W}^T X 1_n}{n} \) as presented in Eq. (13). Solving the second equation of Eq. (15), we have

\[
X \left( X^T \tilde{W} + \frac{1}{n} 1_n 1_n^T - R - \frac{1}{n} 1_n 1_n^T X^T \tilde{W}E \right) + \frac{\lambda}{\alpha} \tilde{W} = 0,
\]

and then get

\[
X \left( 1_n - \frac{1}{n} 1_n 1_n^T \right) X^T \tilde{W} - X \left( 1_n - \frac{1}{n} 1_n 1_n^T \right) R + \frac{\lambda}{\alpha} \tilde{W} = 0,
\]

where we denote the regression target \( (Y + Y \odot M) = R \). Thereby we can derive

\[
\tilde{W} = (XNX^T + (\lambda/\alpha)I_d)^{-1}XNR.
\]

Considering that the weight parameter \( \alpha \) here is essentially a set of fixed values corresponding to each feature in matrix \( X \), we define a diagonal matrix \( A \) to reformulate \( (I_d/\alpha) \).

Thus the optimal solution \( \tilde{W} \) is derived as presented in Eq. (12). Accordingly, we finish the proof. \( \square \)

Following Theorem 1, parameters \( \tilde{W} \) and \( b \) can be updated with the optimal closed-form solutions in Eqs. (12) and (13) within each iteration.

### 4.2. Optimize the adjustment matrix \( M \)

We present the updating rule of the adjustment matrix \( M \) during each iteration based on the following theorem.

**Theorem 2.** Given fixed parameters \( \tilde{W} \) and \( b \), the optimal solution of problem (11) with respect to \( M \) can be calculated as

\[
M = \frac{1}{2} \left( \text{abs}(Y \odot \tilde{V} - \hat{1}_{n \times c}) + (Y \odot \tilde{V} - \hat{1}_{n \times c}) \right),
\]

where we define \( \tilde{V} = X^T \tilde{W} + b \) recording the predicted labels, \( \hat{1}_{n \times c} \) is an \( n \times c \) all-one matrix, and abs(\cdot) denotes the operator which takes the element-wise absolute values of a matrix.

**Proof.** Since parameters \( \tilde{W} \) and \( b \) are fixed, optimization problem (11) can be rewritten as

\[
\min_M \left\| X^T \tilde{W} + b 1_n 1_n^T - Y \odot M \right\|_F^2 \quad \text{s.t.} \quad M \geq 0.
\]

Based on the Frobenius norm theory, problem (21) can be decoupled element-wisely into \( n \times c \) subproblems. For the \( i \)th row and \( j \)th column elements \( \tilde{V}_{ij} \), \( Y_{ij} \), and \( M_{ij} \) respectively in matrices \( \tilde{V}, Y, \) and \( M \in \mathbb{R}^{n \times c}, \) problem (21) can be reformulated as

\[
\min_{M_{ij}} \left( \tilde{V}_{ij} - Y_{ij} - M_{ij} \right)^2, \quad \text{s.t.} \quad M_{ij} \geq 0.
\]

Based on the fact that \( Y \) consists of “± 1” elements only, multiplied by \( Y_{ij}^2 = 1 \), the objective function in problem (22) then becomes \( (Y_{ij} \tilde{V}_{ij} - 1 - M_{ij})^2 \). In consideration of the non-negativity of \( M_{ij} \), we can derive the optimal \( M_{ij} \) as:

\[
M_{ij} = \begin{cases} 
0, & \text{if } Y_{ij} \tilde{V}_{ij} - 1 < 0; \\
Y_{ij} \tilde{V}_{ij} - 1, & \text{otherwise}.
\end{cases}
\]

which can be further formulated as Eq. (20). Hence we finish the proof. \( \square \)

Following Theorem 2, the adjustment matrix \( M \) can be updated with the optimal solution denoted in Eq. (20) during each iteration.

### 4.3. Optimize the adaptive weight parameter \( \alpha \)

We formulate the updating rule of the adaptive weight parameter \( \alpha \) as the following theorem.

**Theorem 3.** Given fixed parameter \( \tilde{W} \), the optimal solution of problem (11) with respect to \( \alpha \) can be derived as

\[
\alpha_k = \frac{\|\tilde{W}_k\|_F}{\sum_{k=1}^v \|\tilde{W}_k\|_F},
\]

where integer \( k \in [1, v] \) is the view index and \( \tilde{W} \) can be denoted as a set of sub-matrices in different views: \( W = [\tilde{W}_1, \tilde{W}_2, \ldots, \tilde{W}_v]^T \).

**Proof.** Since parameter \( \tilde{W} \) is fixed, the optimization problem (11) can be rewritten as

\[
\min_{\alpha} \frac{\lambda}{\sum_{k=1}^v \|\tilde{W}_k\|_F^2}, \quad \text{s.t.} \quad \sum_{k=1}^v \alpha_k = 1, \quad \forall \alpha_k > 0.
\]

Based on Cauchy inequality theory, we have

\[
\sum_{k=1}^v \frac{1}{\alpha_k} \left\| \tilde{W}_k \right\|_F^2 \geq \left( \sum_{k=1}^v \left\| \tilde{W}_k \right\|_F^2 \right)^2,
\]

in which “=” is satisfied if and only if \( \alpha \) takes the set of values \( \{\alpha_k\}_{k=1}^v \), where \( \alpha_k = \left( \|\tilde{W}_k\|_F / \sum_{k=1}^v \|\tilde{W}_k\|_F \right) \). By solving problem (25), we can thereby derive the optimal solution \( \alpha \) presented in Eq. (24). Hence the proof is completed. \( \square \)
Following Theorem 3, the parameter $\alpha$ can be updated with the optimal closed-form solution stated in Eq. (24) within each iteration. Additionally, for $v$ different views, we have $v$ different weight parameters $\{\alpha_i\}_{i=1}^v$, which can be further applied to calculate the new transformation matrix $\tilde{W}$ for the next iteration.

The entire learning procedure is summarized in Algorithm 1.

Algorithm 1 The learning procedure of the proposed method.

Require:
- $n$-sample matrix $\{X_i\}_{i=1}^n$ with $v$ different views;
- Corresponding labels $\{y_i\}_{i=1}^n \in [1, 2, \ldots, c]$ with $c$ different classes;
- Trade-off parameter $\lambda$;
- Termination parameter $\epsilon$ (usually set to $10^{-3}$);
- Maximum number of iterations $t$.

1: **Initialization:**
   $M = 0$, $\alpha_0 = 1/v$ (\forall k), $\tilde{W_0} = 0$, and $b_0 = 0$;
   $X = [X_1, X_2, \ldots, X_v]^T \in \mathbb{R}^{d \times n}$, $\mathbf{Y} \in \mathbb{R}^{n \times c}$ is filled with “$\pm 1$” (see Section 3), and $a = a_0 = [\alpha_1, \alpha_2, \ldots, \alpha_v]^T$;

2: for $j = 1$; $j < t$; $j =$ do
3: Update $\tilde{W}$ and $b$ by Eq. (12) and (13) respectively;
4: Update $M$ by Eq. (20);
5: Update $\alpha_j$ by Eq. (24) ($\mathbf{a}$ is updated simultaneously);
6: if $||\tilde{W} - \tilde{W_0}||_F^2 + ||b - b_0||_2^2 + ||a - a_0||_2^2 < \epsilon$ then
7: break;
8: end if
9: $\tilde{W_0} = \tilde{W}$, $b_0 = b$, $a_0 = a$;
10: end for

Ensure: Transformation matrix $\tilde{W}$ and intercept vector $b$.

Benefiting from the optimal form of each solution, Algorithm 1 can be performed robustly with remarkable convergence speed in extensive experiments.

4.4. Convergence analysis

To analyze the convergence of Algorithm 1, we employ the objective function $f(\tilde{W}, b, M, \alpha)$ in Eq. (14) with parameters $M$ and $\alpha$ no longer fixed. Then we present the following theorem.

**Theorem 4.** Algorithm 1 monotonically decreases the value of $f(\tilde{W}, b, M, \alpha)$ within each iteration until convergence.

**Proof.** For simplicity, we denote the objective function during the $j$th iteration as $f(\tilde{W}_j, b_j, M_j, \alpha_j)$. As presented in Algorithm 1, parameters $\tilde{W}$ and $b$ are updated by $(\tilde{W}_{j+1}, b_{j+1}) = \arg\min_{\tilde{W}, b} f(\tilde{W}, b, M, \alpha)$ with $M$ and $\alpha$ fixed. Considering that the above minimization objective is convex, we naturally have $f(\tilde{W}_{j+1}, b_{j+1}, M_j, \alpha_j) \leq f(\tilde{W}_j, b_j, M_j, \alpha_j)$.

Accordingly, parameter $M$ is updated as $M_{j+1} = \arg\min_M f(\tilde{W}_j, b_j, M, \alpha)$, in which $\tilde{W}_{j+1}$, $b_{j+1}$, and $\alpha_j$ are fixed. The above minimization problem can be referred to Eq. (21). Similarly, we have $f(\tilde{W}_{j+1}, b_{j+1}, M_{j+1}, \alpha_j) \leq f(\tilde{W}_j, b_j, M_j, \alpha_j)$.

For the updating of $\alpha$ during the $(j + 1)$th iteration, we have $\alpha_{j+1} = \arg\min_\alpha f(\tilde{W}_j, b_j, M_j, \alpha)$ with $\tilde{W}_{j+1}$, $b_{j+1}$, and $M_{j+1}$ fixed. Hence we get $f(\tilde{W}_{j+1}, b_{j+1}, M_{j+1}, \alpha_{j+1}) \leq f(\tilde{W}_j, b_j, M_j, \alpha_j)$.

Following the three derived inequalities (27), (28), and (29), we have $f(\tilde{W}_{j+1}, b_{j+1}, M_{j+1}, \alpha_{j+1}) \leq f(\tilde{W}_j, b_j, M_j, \alpha_j)$.

Considering that there is a lower bound of objective function $f(\tilde{W}, b, M, \alpha)$, we can finally draw the conclusion that Algorithm 1 monotonically decreases the value of the objective function within each iteration until convergence. Thus the proof is completed. □

5. Experiments

In this section, we first introduce the datasets in which the experiments are conducted, and then describe the experimental settings and results of the proposed method as well as several comparisons with some important observations.

5.1. Dataset descriptions

Ten benchmark datasets with diverse image types are adopted for evaluations, including Handwritten (HW) [29], Caltech101-7/20 [30], NUS-WIDE-OBJECT [31], Animals with attributes (AwA) [32], SUN397.
**Table 3**
Average learned weights for each individual view using the proposed method.

<table>
<thead>
<tr>
<th>View #</th>
<th>Handwritten</th>
<th>Caltech-7</th>
<th>Caltech-20</th>
<th>NUS-WIDE</th>
<th>AwA</th>
<th>COIL20</th>
<th>COIL100</th>
<th>Scene15</th>
<th>GHIM20</th>
<th>SUN397</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.493</td>
<td>0.0254</td>
<td>0.0167</td>
<td>0.0686</td>
<td>0.0254</td>
<td>0.0035</td>
<td>0.6384</td>
<td>0.1415</td>
<td>0.0031</td>
<td>0.2353</td>
</tr>
<tr>
<td>2</td>
<td>0.0000</td>
<td>0.0605</td>
<td>0.0484</td>
<td>0.1831</td>
<td>0.0605</td>
<td>0.3200</td>
<td>0.0001</td>
<td>0.2686</td>
<td>0.0000</td>
<td>0.1403</td>
</tr>
<tr>
<td>3</td>
<td>0.2320</td>
<td>0.0514</td>
<td>0.0427</td>
<td>0.2699</td>
<td>0.0514</td>
<td>0.0484</td>
<td>0.0571</td>
<td>0.4345</td>
<td>0.0000</td>
<td>0.2534</td>
</tr>
<tr>
<td>4</td>
<td>0.0341</td>
<td>0.8626</td>
<td>0.8922</td>
<td>0.0996</td>
<td>0.8626</td>
<td>0.6170</td>
<td>0.0754</td>
<td>0.1554</td>
<td>0.1067</td>
<td>0.3710</td>
</tr>
<tr>
<td>5</td>
<td>0.1777</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.3788</td>
<td>0.0000</td>
<td>0.0112</td>
<td>0.2289</td>
<td>0.0000</td>
<td>0.0000</td>
<td>–</td>
</tr>
<tr>
<td>6</td>
<td>0.1369</td>
<td>0.0000</td>
<td>0.0000</td>
<td>–</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0992</td>
</tr>
</tbody>
</table>

**Handwritten** contains 2000 images of 10 digit classes from “0” to “9” for handwritten digits recognition. Six published features are adopted for the subsequent evaluations, i.e., Pixel Averages in 2 × 3 Windows (P2W), Fourier Coefficients of the Character Shapes (FOU), Profile Correlations (FAC), Zernike Moment (ZER), Karhunen-Loève Coefficients (KAC), and Morphological (MOR) features.

**Caltech101** consists of 101 classes of images for objective recognition. Due to the unbalance of the data quantity in each class of Caltech101, we follow the previous work [38] and select the commonly-used 7 categories for the subsequent evaluations, including Face, Motorbikes, Doll-Ball, Garfield, Snoopy, Stop-Sign, and Windsor-Chair with 1474 images in total, referred to as Caltech1-7. Moreover, a larger subset of Caltech101 is also involved for the evaluations, namely Caltech1-20, which contains totally 2386 images of 20 categories, i.e., Face, Leopards, Motorbikes, Binocular, Brain, Camera, Car-Side, Doll-Ball, Ferry, Garfield, Hedgehog, Pagoda, Rhino, Snoopy, Stapler, Stop-Sign, Water-Lily, Windsor-Chair, Wrench, and Yin-yang. Five published features are adopted for the experiments, i.e., Gabor, Wavelet Moments (WM), CENTRIST [39], HOOG [40], RIST [1], and LBP [2].

**NUS-WIDE-OBJECT** contains 30,000 real-world object images of 31 categories for objective recognition and image retrieval, which is a subset of the NUS-WIDE dataset (“NUS-WIDE” will be short for “NUS-WIDE-OBJECT” for brevity in this paper). Five officially provided features are adopted for the subsequent evaluations, i.e., Color Histogram (CH), block-wise Color Moments (CM), Color Correlogram (CoR), Edge Direction Histogram (EDH), and Wavelet Texture (WT).

**AwA** contains 30,475 images of 50 kinds of animals for animal recognition and transfer learning. Due to the unbalance of the data quantity in each class, we randomly sample 80 images for each animal class and involve 4000 images for the subsequent evaluations in total. Six officially provided features are adopted, i.e., Global Color Histogram (CQ), Local Self-Similarity (LSS) [40], Pyramid HOG (PHOG) [41], SIFT [3], RMSIFT [42], and SURF [43].

**COIL20/100** are two image datasets respectively composed of 20 and 100 categories, with 72 images in each category. A fea-

ture extraction toolbox\textsuperscript{2} [44–46] is employed for generating six types of features in COIL20/100 datasets, including Color [5,6], GIST [1], HOG\_2 \times 2, HOG\_3 \times 3, LBP [2], and SIFT [3]. Since images in COIL20 are all gray-scale, the “Color” features will not be used in the experiments associated.

Scene15 contains 4485 images of 15 natural scene categories. The features are extracted using the same setting as on COIL20/100, and similarly, the “Color” features are ignored since Scene15 contains only gray-scale images.

GHIM20 contains 10,000 images of 20 diverse categories for image retrieval. Five types of features, i.e., Color [5,6], GIST [1], HOG\_2 \times 2, LBP [2], and SIFT [3], are extracted using the toolbox mentioned above. We have also involved another deep learning based feature type that is taken from the last average pooling layer (“cls3\_pool”) of GoogLeNet [47] (pretrained on the ImageNet ILSVRC challenge dataset). MatConvNet [48] toolbox is employed for the deep learning based feature extraction.

SUN397 contains 108,754 images of 397 scene categories for large-scale scene understanding. We have extracted four types of deep learning based features, which are respectively from the last pooling layer (“pool5”) of ResNet-50 [49], the last pooling layer (“cls3\_pool”) of GoogLeNet [47], the penultimate fully connected layer (“fc7”) of VGG-16 [50], and the penultimate fully connected layer (“fc7”) of AlexNet [51]. The features are extracted using MatConvNet [48], and all the four pretrained (on ImageNet dataset) deep learning models are obtained at MatConvNet homepage\textsuperscript{3}.

\subsection{5.2. Experimental setup}

We compare our proposed method with five state-of-the-art methods, including a conventional single-view classification method, i.e., SVM [52], a supervised dimension reduction method, i.e., LDA [53], and three supervised multi-view learning methods, i.e., SimpleMKL [21], Multi-View Matrix Decomposition (MVMD) [54], and Multi-View Correlated feature learning with Shared Component (MVCS) [55].

We apply SVM on each individual view to assess the single view performance. In contrast, the results of SVM applied on the concatenation of all views are taken as the baselines. The proposed method and LDA are respectively evaluated with a $k$-NN classifier based on the features in the projected subspace. We also adopt the original concatenated features for comparison. Since different views usually yield different performances, the top three best-performed views of each dataset (measured on the single-view SVM evaluations) are concatenated as each “best view combination”, where we apply SVM again to supplement the evalua-\textsuperscript{2} \url{https://github.com/adikhosla/feature-extraction/}
\textsuperscript{3} \url{http://www.vlfeat.org/matconvnet/}
In order to further evaluate the effectiveness of the learned adaptive weights, we apply SVM and $k$-NN on the concatenated features weighted by the learned parameters respectively. Incidentally, we choose several appropriate values of parameter $k$ in terms of the ten different datasets in the experiments, i.e., $k = 7$ for the datasets of NUS-WIDE, AwA, Scene15 and SUN397; and $k = 1$ for all the rest.

We conduct standard 5-fold cross-validation and report the average results on each dataset. Within the training data for each of the 5 trials, an internal 5-fold cross-validation is performed to fine-tune the parameters. Specifically, the trade-off parameter ($\lambda$ in Eq. (11)) of our proposed method is fine-tuned by searching from the range of $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. As for SVM and SimpleMKL, one Gaussian kernel (RBF) is constructed for each view of the datasets, i.e., $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|_2^2)$, where parameter $\gamma$ is selected in the range of $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. The trade-off parameters of the two methods are fine-tuned in the same range used in our method. LibSVM [56] software package is employed for implementations. As for LDA, MVMD and MVCS, we follow the default experimental settings.

Classification accuracy serves as the evaluation metric for all the experiments, which is depicted as

$$\text{Accuracy} = \frac{\sum_{i=1}^{n} \delta(y_i, \hat{y}_i)}{n},$$

where $y_i$ is the ground truth of each label, $\hat{y}_i$ is the corresponding predicted label, and $n$ is the number of test samples. The function $\delta(\cdot, \cdot)$ measures the two input arguments, and outputs “1” if the two arguments are equal; or outputs “0” otherwise.
5.3. Results and analysis

We list the average learned weights of individual views per each dataset in Table 3 with our proposed method. Table 4 collects the classification accuracy and the standard deviation obtained with our proposed method and the competitors on ten datasets. We note that the original concatenated features are denoted as “All views”, the top three best-performed views are concatenated as “Top 3 views”, and the concatenated features weighted by the learned adaptive weights are marked with “Weighted”. As listed in Table 4, our proposed method consistently outperforms all the other compared competitors, demonstrating the superiority of our method. Moreover, the methods that utilize multiple feature types generally perform better than SVM using each single view, verifying the rationality of the fusion of multiple views. However, applying SVM on some specific views can sometimes achieve better performance than on the concatenated one, confirming that the direct feature concatenation of multiple views is not an optimal solution. The results of applying SVM on the concatenation of top three best-performed views are generally better than any individual views (as well as all concatenated views), inferring that rational combination of different views can usually bring a certain boost on performance. In contrast, as we integrate the multi-view features with the learned weights, the performances of both SVM and k-NN meet another certain improvement, suggesting the effectiveness of the adaptive-weighting strategy. Our proposed method yields better results than all the compared multi-view learning approaches, validating the importance of preserving and fusing the correlative and the complementary information. As we compare the learned weights in Table 3 with the single-view SVM classification accuracy in Table 4, we can tell that the high-weighted views generally yield relatively good performances. However, this pattern is not followed by every view in all the datasets involved, because our method can achieve an optimal fusion to make the most of the correlative and the complementary information, but barely determine which view offers more boost on accuracy. Additionally, benefiting from the discriminative low-dimensional features we learned, our proposed method can thus perform superbly with a simple classifier. Again, the above important observations demonstrate the advantages of the proposed method.

We report the performance of the proposed method on the ten datasets with the trade-off parameter $\lambda$ varying from the range of $(10^{-3}, 10^{-2}, \ldots, 10^{2}, 10^{3})$ in Fig. 2. Different values of $\lambda$ lead to different weights we learn. Typically, the optimal weights are highly data-dependent, but we can roughly observe that the range of [0.1,10] for $\lambda$ usually generates satisfactory results.

Now we consider the computational complexity of our proposed method. Specifically, the computational complexity of step 3, 4 and 5 in Algorithm 1 respectively scales in $O(nd^2)$, $O(ndc)$ and $O(dc)$. Considering that the feature dimension across all views is always larger than the class number, the computational complexity of Algorithm 1 scales in $O(ndc)$. Table 5 collects the training time of our proposed method and five competitors on the ten datasets. Algorithms are tested on an x64-based PC with 8 processors (3.60 GHz for each) and 32.0 GB RAM memory using MATLAB R2016b (9.1.0). Generally speaking, our method performs stably and exhibits a promising training speed compared with the competitors, especially on the large-scale datasets. Fig. 3 illustrates the convergence curves in terms of the objective function values on the ten datasets when the trade-off parameter $\lambda$ is set to 1. Overall, the convergence property varies with datasets and different trade-off parameters. Empirically, it often takes more iterations to converge when the trade-off parameter is set too large or too small, and the minimal number of iterations required usually occurs when the best classification performance is achieved.

6. Conclusion

In this paper, we present a novel supervised multi-view feature learning framework, which is able to capture the correlative and the complementary information with a discriminative technique. The proposed framework is formulated in a regression-based structure with the introductions of a new discriminative regression target and a set of learnable weights. Applying the learned adaptive-weighting transformation on the concatenation of the original multi-view features, the correlative and the complementary information can be preserved in a low-dimensional discriminative subspace, which yields promising results in the subsequent classification tasks. An efficient iterative algorithm is exploited to optimize the proposed framework with the optimal closed-form solutions. Comparisons with the existing state-of-the-art competitors on ten broadly-used benchmark datasets demonstrate our superiority in classification and efficacy for high-dimensional data in the supervised multi-view learning scenario.

The main restriction of the proposed method is the relatively long training time. The optimization procedure comprises three alternate iterative steps for four parameters to be estimated. Although the convergence of the algorithm is guaranteed, it can sometimes take too much time to converge. As future work, we intend to speed up our algorithm with efforts into different initialization and optimization techniques. We would also like to extend our method to the semi-supervised scenario.

Acknowledgments

Our work was supported in part by the National Natural Science Foundation of China under Grant 61572388 and 61703327, in part by the Key R&D Program—The Key Industry Innovation Chain of Shaanxi under Grant 2017ZDCXL-GY-05-04-02 and 2017ZDCXL-GY-05-02.

References


Muli Yang is now a Ph.D. student at the School of Electronic Engineering, Xidian University, Xi’an, China. His research interest includes pattern recognition, computer vision and machine learning.

Cheng Deng (S’09) received the B.E., M.S., and Ph.D. degrees in signal and information processing from Xidian University, Xi’an, China. He is currently a Full Professor with the School of Electronic Engineering at Xidian University. His research interests include computer vision, multimedia processing and analysis, and information hiding. He is the author and coauthor of more than 50 scientific articles at top venues, including IEEE TNNLS, TMM, TCYB, TSMC, TP, ICCV, CVPR, IJCAI, and AAAI.

Feiping Nie received the Ph.D. degree in Computer Science from Tsinghua University, China in 2009, and currently is a Full Professor in Northwestern Polytechnical University, Xi’an, China. His research interests are machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing and information retrieval. He has published more than 100 papers in the following journals and conferences: TPAMI, IJCV, TIP, TNNLS, TKDE, ICML, NIPS, KDD, IJCAI, AAAI, ICCV, CVPR, ACM MM. His papers have been cited more than 10000 times and the H-index is 53. He is now serving as Associate Editor or PC member for several prestigious journals and conferences in the related fields.