Multi-view Semantic Learning for Data Representation

Peng Luo[1], Jinye Peng[1](✉), Ziyu Guan[1], and Jianping Fan[2]

[1] College of the Information and Technology,
Northwest University of China, Xi'an, China
luopengpeng@gmail.com, {pjy,ziyuguan}@nwu.edu.cn
[2] Department of Computer Science,
University of North Carolina at Charlotte, Charlotte, NC, USA
jfan@uncc.edu

Abstract. Many real-world datasets are represented by multiple features or modalities which often provide compatible and complementary information to each other. In order to obtain a good data representation that synthesizes multiple features, researchers have proposed different multi-view subspace learning algorithms. Although label information has been exploited for guiding multi-view subspace learning, previous approaches either fail to directly capture the semantic relations between labeled items or unrealistically make Gaussian assumption about data distribution. In this paper, we propose a new multi-view nonnegative subspace learning algorithm called Multi-view Semantic Learning (MvSL). MvSL tries to capture the semantic structure of multi-view data by a novel graph embedding framework. The key idea is to let neighboring intra-class items be near each other while keep nearest inter-class items away from each other in the learned common subspace across multiple views. This nonparametric scheme can better model non-Gaussian data. To assess nearest neighbors in the multi-view context, we develop a multiple kernel learning method for obtaining an optimal kernel combination from multiple features. In addition, we encourage each latent dimension to be associated with a subset of views via sparseness constraints. In this way, MvSL is able to capture flexible conceptual patterns hidden in multi-view features. Experiments on two real-world datasets demonstrate the effectiveness of the proposed algorithm.

Keywords: Multi-view learning · Nonnegative matrix factorization · Graph embedding · Multiple kernel lerning · Structured sparsity

1 Introduction

In many real-world data analytic problems, instances are often described with multiple modalities or views. It becomes natural to integrate multi-view representations to obtain better performance than relying on a single view. A good

A. Appice et al. (Eds.): ECML PKDD 2015, Part I, LNAI 9284, pp. 367–382, 2015.
DOI: 10.1007/978-3-319-23528-8_23

integration of multi-view features can lead to a more comprehensive description of the data items, which could improve performance of many related applications.

An emerging area of multi-view learning is *multi-view latent subspace learning*, which aims to obtain a compact latent representation by taking advantage of inherent structure and relation across multiple views. A pioneering technique in this area is Canonical Correlation Analysis (CCA) [7], which tries to learn the projections of two views so that the correlation between them is maximized. Recently, a lot of methods have been applied to multi-view subspace learning, such as matrix factorization [9], [4], [11], [18], graphical models [3] and spectral embedding [22].

Matrix factorization techniques have received more and more attention as fundamental tools for multi-view latent subspace learning. Since a useful representation acquired by matrix factorization typically makes latent structure in the data explicit (through the basis vectors), and usually reduces the dimensionality of input views, so that further analysis can be effectively and efficiently carried out. Nonnegative Matrix Factorization (NMF) [13] is an attractive matrix factorization method due to its theoretical interpretation and desired performance. NMF aims to find two nonnegative matrices (a basis matrix and an encoding matrix) whose product provides a good approximation to the original matrix. NMF tries to formulate a feasible model for learning object parts, which is closely relevant to human perception mechanism. Recently, variants of NMF have been proposed for multi-view learning [11], [18], [10].

Labeled data has been incorporated to improve NMF's performance in both the single view case [16], [21] and the multi-view case [10]. However, there is still lack of effective methods for learning a common nonnegative latent subspace which captures the semantic structure of multi-view data through label information. Previously, there are mainly two ways to incorporate label information into the NMF framework. The first one is to reconstruct the label indicator matrix through multiplying the encoding matrix by a weight matrix [10], [17],[16]. These methods intrinsically impose *indirect* affinity constraints on encodings of labeled items. Nevertheless, such indirect constraints could be insufficient for capturing the semantic relationships between labeled items. The second one is to regularize the encodings of labeled items by fisher-style discriminative constraints [21], [25]. Although methods of this kind directly penalize distances among labeled items in the latent subspace, they assume the data of each class follows a Gaussian distribution. However, in reality this assumption is too restricted since data often exhibit complex non-Gaussian distribution [2], [24].

In this paper, we propose a novel semi-supervised multi-view representation (i.e. latent subspace) learning algorithm, namely, Multi-view Semantic Learning (MvSL), to better capture the semantic structure of multi-view data. MvSL jointly factorizes data matrices of different views, and each view is factorized into a *basis matrix* and an *encoding matrix* where the encoding matrix is the low dimensional optimal consensus representation shared by multiple views. We regularize the encoding matrix by developing a novel graph embedding framework:

we construct (1) an affinity graph which characterizes the intra-class compactness and connects each data point with its neighboring points of the same class; (2) a discrimination graph which connects the marginal points and characterizes the inter-class separability in the learned subspace. This nonparametric scheme can better capture the complex non-Gaussian distribution of real-world data [24]. A sub-challenge is how to identify nearest neighbors in the multi-view context. To this end, we develop a new multiple kernel learning algorithm to find the optimal kernel combination for multi-view features. The algorithm tries to optimally preserves the semantic relations among labeled items, so that we can assess within-class variance and between-class similarity effectively. Moreover, we impose a $L_{1,2}$ norm regularizer on the basis matrix to encourage some basis vectors to be zero-valued [9]. In this way, each latent dimension has the flexibility to be associated with a subset of views, thus enhancing the expressive power of the model. To solve MvSL, we develop a block coordinate descent [15] optimization algorithm. For empirical evaluation, two real-world multi-view datasets are employed. The encouraging results of MvSL are achieved in comparison with the state-of-the-art algorithms.

2 Related Work

In this section, we present a brief review of related work about NMF-based subspace learning. Firstly, we describe the notations used throughout the paper.

2.1 Common Notations

In this paper, vectors and matrices are denoted by lowercase boldface letters and uppercase boldface letters respectively. For matrix $\mathbf{M}$, we denote its (i, j)-th element by M_{ij}. The i-th element of a vector $\mathbf{b}$ is denoted by b_i. Given a set of N items, we use matrix $\mathbf{X} \in \mathbb{R}_+^{M \times N}$ to denote the nonnegative data matrix where the i-th column vector is the feature vector for the i-th item. In the multi-view setting, we have H views and the data matrix of the v-th view is denoted by $\mathbf{X}^{(v)} \in \mathbb{R}_+^{M_v \times N}$, where M_v is the dimensionality of the v-th view. Throughout this paper, $\|\mathbf{M}\|_F$ denotes the Frobenius norm of matrix $\mathbf{M}$.

2.2 NMF-Based Latent Subspace Learning

NMF is an effective subspace learning method to capture the underlying structure of the data in the parts-based low dimensional representation space, which accords with the cognitive process of human brain from the psychological and physiological studies [13].

Given an input nonnegative data matrix $\mathbf{X} \in \mathbb{R}_+^{M \times N}$ where each column represents a data point and each row represents a feature. NMF aims to find two nonnegative matrices $\mathbf{U} \in \mathbb{R}_+^{M \times K}$ and $\mathbf{V} \in \mathbb{R}_+^{K \times N}$ whose product can well approximate the original data matrix:

$$\mathbf{X} \approx \mathbf{UV}.$$

$K < \min(M, N)$ denotes the desired reduced dimensionality, and to facilitate discussion, we call $\mathbf{U}$ the basis matrix and $\mathbf{V}$ the coefficient matrix.

It is known that the objective function above is not convex in $\mathbf{U}$ and $\mathbf{V}$ together, so it is unrealistic to expect an algorithm to find the global minimum. Lee and Seung [13] presented multiplicative update rules to find the locally optimal solution as follows:

$$\begin{aligned} U_{ik}^{t+1} =& U_{ik}^{t} \frac{(\mathbf{X}(\mathbf{V}^t)^T)_{ik}}{(\mathbf{U}^t\mathbf{V}^t(\mathbf{V}^t)^T)_{ik}} \\ V_{kj}^{t+1} =& V_{kj}^{t} \frac{((\mathbf{U}^{t+1})^T\mathbf{X})_{kj}}{((\mathbf{U}^{t+1})^T\mathbf{U}^{t+1}\mathbf{V}^t)_{kj}}. \end{aligned}$$

In recent years, many variants of the basic NMF model have been proposed. We just list a few which are related to our work. One direction related to our work is coupling label information to NMF [21], [25]. These works added discriminative constraints into NMF via regularizing the encoding matrix $\mathbf{V}$ by fisher-style discriminative constraints. Nevertheless, fisher discriminative analysis assumes data of each class is approximately Gaussian distributed, a property that cannot always be satisfied in real-world applications. Our method adopts a nonparametric regularization scheme (i.e. regularization in neighborhoods) and consequently can better model real-life data. Another related direction of NMF is sparse NMF [8]. Sparseness constraints not only encourage local and compact representations, but also improve the stability of the decomposition. Most previous works on sparse NMF employed L_1 norm or ratio between L_1 norm and L_2 norm to achieve sparsity on $\mathbf{U}$ and $\mathbf{V}$. However, the story for our problem is different since we have multiple views and the goal is to allow each latent dimension to be associated with a subset of views. Therefore, $L_{1,2}$ norm [9] is used to achieve this goal.

There are also some extensions of NMF for multi-view data, e.g. clustering [18], image annotation [11], graph regularized multi-view NMF [6] and semi-supervised learning [10],[17]. Although [10] and [17] also exploited label information, they incorporated label information as a factorization constraint on $\mathbf{V}$, i.e. reconstructing the label indicator matrix through multiplying $\mathbf{V}$ by a weight matrix. Hence, those methods intrinsically imposed *indirect* affinity constraints on encodings of labeled items in the latent subspace. On the contrary, our method *directly* captures the semantic relationships between items in the latent subspace through the proposed graph embedding framework. We will compare MvSL with [10] in experiments.

3 Multi-view Semantic Learning

In this section, we present the proposed Multi-view Semantic Learning (MvSL) algorithm for latent representation learning from multi-view data.

3.1 Matrix Factorization with Multi-view Data

The consensus principle is the fundamental principle in multi-view learning [23]. At first, MvSL jointly factorizes $\{\mathbf{X}^{(v)}\}_{v=1}^{H}$ with different basis matrices $\{\mathbf{U}^{(v)}\}_{v=1}^{H}$ and the consensus encoding matrix $\mathbf{V}$ [9], [18], [11]:

$$\begin{aligned} \min_{\{\mathbf{U}^{(v)}\}_{v=1}^{H},\mathbf{V}} \quad & \frac{1}{2}\sum_{v=1}^{H}\|\mathbf{X}^{(v)} - \mathbf{U}^{(v)}\mathbf{V}\|_F^2 \\ \text{s.t.} \quad & U_{ik}^{(v)} \geq 0, V_{kj} \geq 0, \ \ \forall i, j, k, v. \end{aligned} \tag{1}$$

However, the standard unsupervised NMF fails to discover the semantic structure in the data. In the next, we introduce our graph embedding framework for multi-view semantic learning.

3.2 Graph Embedding for Multi-view Semantic Learning

Let $\mathbf{V}^l \in \mathbb{R}^{K\times N^l}$, the first N^l columns of $\mathbf{V}$, be the latent representation of the first N^l labeled items and $\mathbf{V}^u \in \mathbb{R}^{K\times N^u}$ be the latent representation of the remaining N^u unlabeled items ($i.e.\mathbf{V} = [\mathbf{V}^l\ \mathbf{V}^u]$). Inspired by [24], we propose a graph embedding framework for capturing the semantic structure of multi-view data. We define an affinity graph G^a and a discrimination graph G^p. The affinity graph $G^a = \{\mathbf{V}^l, \mathbf{W}^a\}$ is an undirected weighted graph with labeled item set $\mathbf{V}^l$ as its vertex set, and similarity matrix $\mathbf{W}^a \in \mathbb{R}^{N^l\times N^l}$ which characterizes the intra-class local similarity structure. The discrimination graph $G^p = \{\mathbf{V}^l, \mathbf{W}^p\}$ characterizes inter-class separability and penalizes the similarity between the most similar inter-class item pairs in the learned subspace. Let $\mathbf{v}_i^l$ be the i-th column of $\mathbf{V}^l$. The graph-preserving criteria are given as follows:

$$\min_{\mathbf{V}^l} \frac{1}{2}\sum_{i=1}^{N^l}\sum_{j=1}^{N^l} W_{ij}^a\|\mathbf{v}_i^l - \mathbf{v}_j^l\|_2^2 = \min_{\mathbf{V}^l} \frac{1}{2} tr\left[\mathbf{V}^l\mathbf{L}^a\left(\mathbf{V}^l\right)^T\right], \tag{2}$$

$$\max_{\mathbf{V}^l} \frac{1}{2}\sum_{i=1}^{N^l}\sum_{j=1}^{N^l} W_{ij}^p\|\mathbf{v}_i^l - \mathbf{v}_j^l\|_2^2 = \max_{\mathbf{V}^l} \frac{1}{2} tr\left[\mathbf{V}^l\mathbf{L}^p\left(\mathbf{V}^l\right)^T\right], \tag{3}$$

where $tr(\cdot)$ denotes the trace of a matrix, and $\mathbf{L}^a = \mathbf{D}^a - \mathbf{W}^a$ is the graph Laplacian matrix for G^a with the (i,i)-th elements of the diagonal matrix $\mathbf{D}^a$ equals $\sum_{j=1}^{N^l} W_{ij}^a$ ($\mathbf{L}^p$ is for G^p). Generally speaking, Eq. (2) means items belonging to the same class should be near each other in the learned latent subspace, while Eq. (3) tries to keep items from different classes as distant as possible. However, only with the nonnegative constraints Eq. (3) would diverge. Note that there is an arbitrary scaling factor in solutions to problem (1): for any invertible $K \times K$ matrix $\mathbf{Q}$, we have $\mathbf{U}^{(v)}\mathbf{V} = (\mathbf{U}^{(v)}\mathbf{Q})(\mathbf{Q}^{-1}\mathbf{V})$. Hence, without loss of generality, we add the constraints $\{V_{kj} \leq 1, \forall k, j\}$ to put an upper bound on (3).

The similarity matrices $\mathbf{W}^a$ and $\mathbf{W}^p$ are defined as follows

$$W_{ij}^a = \begin{cases} 1, & \text{if } i \in N_{k_a}(j) \;\; \mathit{or} \;\; j \in N_{k_a}(i) \\ 0, & \text{otherwise} \end{cases}, \tag{4}$$

where $N_{k_a}(i)$ indicates the index set of the k_a nearest neighbors of item i in the *same* class,

$$W_{ij}^p = \begin{cases} 1, & \text{if } i \in N_{k_p}(j) \;\; \mathit{or} \;\; j \in N_{k_p}(i) \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$

where $N_{k_p}(i)$ indicates the index set of the k_p nearest neighbors of item i in the *distinct* classes. We can see from the definitions of $\mathbf{W}^a$ and $\mathbf{W}^p$ that G^a and G^p intrinsically preserve item semantic relations in local neighborhoods. This nonparametric scheme can better handle real-life datasets which often exhibit non-Gaussian distribution.

The remaining question is how to estimate nearest neighbors, which is a routine function for constructing G^a and G^p. However, since real-life datasets could be diverse and noisy, a single feature may not be sufficient to characterize the affinity relations among items. Hence, we propose to use multiple features for assessing the similarity between data items. In particular, we develop a novel Multiple Kernel Learning (MKL) [20],[5] method for this task.

3.2.1 Multiple Kernel Learning

A kernel function measures the similarity between items in terms of one view. We use $\mathbf{K}_v(i,j)$ to denote the kernel value between items i and j in terms of view v. To make all kernel functions comparable, we normalize each kernel function into $[0,1]$ as follows:

$$\mathbf{K}_v(i,j) \leftarrow \frac{\mathbf{K}_v(i,j)}{\sqrt{\mathbf{K}_v(i,i)\mathbf{K}_v(j,j)}}. \tag{6}$$

To obtain a comprehensive kernel function, we linearly combine multiple kernels as follow:

$$\mathbf{K}(i,j,\boldsymbol{\eta}) = \sum_{v=1}^{H} \eta_v \mathbf{K}_v(i,j), \quad \sum_{v=1}^{H} \eta_v = 1, \eta_v \geq 0, \tag{7}$$

where $\boldsymbol{\eta} = [\eta_1, ..., \eta_H]^T$ is the weight vector to be learned. This combined kernel function can lead to better estimation of similarity among items than any single kernel. For example, only relying on color information could not handel images of concept “zebra” well since the background may change arbitrarily, while adding texture information can better characterize zebra images.

Then we need to design the criterion for learning $\boldsymbol{\eta}$. Since our goal is to model the semantic relations among items, the learned kernel function should be accommodated to the semantic structure among classes. We define an *ideal kernel* to encode the semantic structure:

$$\mathbf{K}_{ideal}(i,j) = \begin{cases} 1, & \text{if } y_i = y_j \\ 0, & \text{otherwise} \end{cases}, \tag{8}$$

where y_i denotes the label of item i. For each pair of items, we require its combined kernel function value to conform to the corresponding ideal kernel value. This leads to the following least square loss

$$l(i,j,\boldsymbol{\eta}) = (\mathbf{K}(i,j,\boldsymbol{\eta}) - \mathbf{K}_{ideal}(i,j))^2 \tag{9}$$

Summing $l(i,j,\boldsymbol{\eta})$ over all pairs of labeled items, we could get the optimization objective. However, in reality we would get imbalanced classes: the numbers of labeled items for different classes can be quite different. The item pairs contributed by classes with much larger number of items will dominate the overall loss. In order to tackle this issue, we normalize the contribution of each pair of classes (including same-class pairs) by its number of item pairs. This is equivalent to multiplying each $l(i,j,\boldsymbol{\eta})$ by a weight t_{ij} which is defined as follows

$$\mathbf{t}_{ij} = \begin{cases} \frac{1}{n_i^2}, & \text{if } y_i = y_j \\ \frac{1}{2n_i n_j}, & \text{otherwise} \end{cases}, \tag{10}$$

where n_i denotes the number of items belonging to the class with label y_i. Therefore, the overall loss becomes $\sum_{i,j} t_{ij} l(i,j,\boldsymbol{\eta})$. To prevent overfitting, a L_2 regularization term is added for $\boldsymbol{\eta}$. The final optimization problem is formulated as

$$\begin{aligned} &\min_{\boldsymbol{\eta}} \sum_{i,j=1}^{N^l} t_{ij} l(i,j,\boldsymbol{\eta}) + \lambda \|\boldsymbol{\eta}\|_2^2 \\ &s.t. \sum_{v=1}^{H} \eta_v = 1, \eta_v \geq 0 \end{aligned} \tag{11}$$

where λ is a regularization tradeoff parameter. The optimization problem of (11) is a classical quadratic programming problem which can be solved efficiently using any convex programming software. When $\boldsymbol{\eta}$ is obtained, we could assess the similarity relationship between labeled items in terms of multi-view features according to (7). Then, according to Eqs. (4) and (5) we can construct the affinity matrix $\mathbf{W}^a$ and discriminative matrix $\mathbf{W}^p$, respectively.

3.3 Sparseness Constraint

Since similarities among data items belonging to the same class share the same sparsity pattern, a structured sparseness regularizer is added to objective function to encourage some basis column vectors in $\mathbf{U}^{(v)}$ to become to 0. This makes view v independent of the latent dimensions which correspond to these zeros-valued basis vectors. By employing $L_{1,q}$ norm regularization, the latent factors obtained by NMF can be improved with an additional property of shared sparsity. In this work, we choose $q = 2$. $L_{1,2}$ norm of matrix $\mathbf{U}$ is defined as:

$$\|\mathbf{U}\|_{1,2} = \sum_{k=1}^{K} \|\mathbf{u}_k\|_2, \tag{12}$$

3.4 Objective Function of MvSL

By synthesizing the above objectives, the optimization problem of MvSL is formulated as:

$$\begin{aligned} \min_{\{\mathbf{U}^{(v)}\}_{v=1}^{H}, \mathbf{V}} \quad & \frac{1}{2}\sum_{v=1}^{H} \|\mathbf{X}^{(v)} - \mathbf{U}^{(v)}\mathbf{V}\|_F^2 + \alpha \sum_{v=1}^{H} \|\mathbf{U}^{(v)}\|_{1,2} \\ & + \frac{\beta}{2}\left\{tr\left[\mathbf{V}^l\mathbf{L}^a(\mathbf{V}^l)^T\right] - tr\left[\mathbf{V}^l\mathbf{L}^p(\mathbf{V}^l)^T\right]\right\} \\ \text{s.t.} \quad & U_{ik}^{(v)} \geq 0, 1 \geq V_{kj} \geq 0, \ \ \forall i, j, k, v. \end{aligned} \tag{13}$$

4 Optimization

The joint optimization function in (13) is not convex over all variables $\mathbf{U}^{(1)}, ..., \mathbf{U}^{(H)}$ and $\mathbf{V}$ simultaneously. Thus, we propose a block coordinate descent method [15] which optimizes one block of variables while keeping the other block fixed. The procedure is depicted in Algorithm 1. For the ease of representation, we define

$$\begin{aligned} \mathcal{O}\{(\mathbf{U}^{(1)}, ..., \mathbf{U}^{(H)}, \mathbf{V})\} = & \frac{1}{2}\sum_{v=1}^{H} \|\mathbf{X}^{(v)} - \mathbf{U}^{(v)}\mathbf{V}\|_F^2 + \alpha \sum_{v=1}^{H} \|\mathbf{U}^{(v)}\|_{1,2} \\ & + \frac{\beta}{2}\left\{tr\left[\mathbf{V}^l\mathbf{L}^a(\mathbf{V}^l)^T\right] - tr\left[\mathbf{V}^l\mathbf{L}^p(\mathbf{V}^l)^T\right]\right\} \end{aligned} \tag{14}$$

4.1 Optimizing $\{\mathbf{U}^{(v)}\}_{v=1}^{H}$

When $\mathbf{V}$ is fixed, $\mathbf{U}^{(1)}, ..., \mathbf{U}^{(H)}$ are independent with one another. Since the optimization method is the same, here we just focus on an arbitrary view and use $\mathbf{X}$ and $\mathbf{U}$ to denote respectively the data matrix and the basis matrix for the view. The optimization problem involving $\mathbf{U}$ can be written as

$$\begin{aligned} \min_{\mathbf{U}} \quad & \phi(\mathbf{U}) := \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2 + \alpha\|\mathbf{U}\|_{1,2} \\ \text{s.t.} \quad & U_{ik} \geq 0, \ \ \forall i, k. \end{aligned} \tag{15}$$

Two terms of $\phi(\mathbf{U})$ are convex functions. The first term of $\phi(\mathbf{U})$ is differentiable, and its gradient is Lipschitz continuous. Hence, an efficient convex optimization method can be adopted. [12] presented a variant of Nesterov's first order method, suitable for solving (15). In this paper, we take the optimization method of [12] to update $\mathbf{U}$. Due to the limitation of space, details can be found in [12].

4.2 Optimizing V

When $\{\mathbf{U}^{(v)}\}_{v=1}^{H}$ are fixed, the subproblem for $\mathbf{V}$ can be written as

$$\begin{aligned} \min_{\mathbf{V}} \; \psi(\mathbf{V}) := \Bigg(& \frac{1}{2}\sum_{v=1}^{H} \|\mathbf{X}^{(v)} - \mathbf{U}^{(v)}\mathbf{V}\|_F^2 \\ & + \frac{\beta}{2}\left\{ tr\left[\mathbf{V}^l\mathbf{L}^a(\mathbf{V}^l)^T\right] - tr\left[\mathbf{V}^l\mathbf{L}^p(\mathbf{V}^l)^T\right]\right\}\Bigg) \\ \text{s.t.} \quad & 1 \geq V_{kj} \geq 0, \;\; \forall j,k. \end{aligned} \tag{16}$$

This is a bounded nonnegative quadratic programming problem for $\mathbf{V}$. Sha *et al.* [19] proposed a general multiplicative optimization scheme for this kind of problems. Inspired by their method, we develop a multiplicative update algorithm for optimizing $\mathbf{V}$.

Firstly, recall that $\mathbf{X}^{(v)} = [\mathbf{X}^{(v),l}\ \mathbf{X}^{(v),u}]$ and $\mathbf{V} = [\mathbf{V}^l\ \mathbf{V}^u]$. We can transform the first term of $\psi(\mathbf{V})$:

$$\begin{aligned} & \frac{1}{2}\sum_{v=1}^{H} \|\mathbf{X}^{(v)} - \mathbf{U}^{(v)}\mathbf{V}\|_F^2 \\ = & \frac{1}{2}\sum_{v=1}^{H}\Big(tr[(\mathbf{V}^l)^T(\mathbf{U}^{(v)})^T\mathbf{U}^{(v)}\mathbf{V}^l] - 2tr[(\mathbf{V}^l)^T(\mathbf{U}^{(v)})^T\mathbf{X}^{(v),l}] \\ & + tr[(\mathbf{V}^u)^T(\mathbf{U}^{(v)})^T\mathbf{U}^{(v)}\mathbf{V}^u] - 2tr[(\mathbf{V}^u)^T(\mathbf{U}^{(v)})^T\mathbf{X}^{(v),u}]\Big) + const. \end{aligned}$$

For convenience, let $\mathbf{P} = \sum_{v=1}^{H}(\mathbf{U}^{(v)})^T\mathbf{U}^{(v)}$ and $\mathbf{Q}^l = \sum_{v=1}^{H}(\mathbf{U}^{(v)})^T\mathbf{X}^{(v),l}$. $\mathbf{Q}^u$ is defined similarly for the unlabeled part. Eq. (16) can be transformed into

$$\begin{aligned} \min_{\mathbf{V}} \; & \frac{1}{2}tr[(\mathbf{V}^l)^T\mathbf{P}\mathbf{V}^l] - tr[(\mathbf{V}^l)^T\mathbf{Q}^l] + \frac{1}{2}tr[(\mathbf{V}^u)^T\mathbf{P}\mathbf{V}^u] - tr[(\mathbf{V}^u)^T\mathbf{Q}^u] \\ & + \frac{\beta}{2}\left\{ tr\left[\mathbf{V}^l\mathbf{L}^a(\mathbf{V}^l)^T\right] - tr\left[\mathbf{V}^l\mathbf{L}^p(\mathbf{V}^l)^T\right]\right\} \\ \text{s.t.} \quad & 1 \geq V_{kj} \geq 0, \;\; \forall j,k. \end{aligned} \tag{17}$$

Since $\mathbf{V}^l$ and $\mathbf{V}^u$ are independent, we analyze them separately. The objective terms involving $\mathbf{V}^l$ can be summarized as

$$\begin{aligned} \mathcal{O}^l(\mathbf{V}^l) = & \frac{1}{2}tr[(\mathbf{V}^l)^T\mathbf{P}\mathbf{V}^l] - tr[(\mathbf{V}^l)^T\mathbf{Q}^l] \\ & + \frac{\beta}{2}\left\{ tr\left[\mathbf{V}^l\mathbf{L}^a(\mathbf{V}^l)^T\right] - tr\left[\mathbf{V}^l\mathbf{L}^p(\mathbf{V}^l)^T\right]\right\} \end{aligned} \tag{18}$$

The second term is linear term for $\mathbf{V}^l$. We only need to focus on the quadratic terms which can be rewritten as follows

$$\frac{1}{2}tr[(\mathbf{V}^l)^T\mathbf{P}\mathbf{V}^l] = \frac{1}{2}\sum_{j=1}^{N^l}(\mathbf{v}_j^l)^T\mathbf{P}\mathbf{v}_j^l, \tag{19}$$

$$\begin{aligned}&\frac{\beta}{2}\left\{tr\left[\mathbf{V}^l\mathbf{L}^a(\mathbf{V}^l)^T\right] - tr\left[\mathbf{V}^l\mathbf{L}^p(\mathbf{V}^l)^T\right]\right\}\\=&\frac{\beta}{2}\sum_{k=1}^{K}\left\{(\bar{\mathbf{v}}_k^l)^T(\mathbf{D}^a+\mathbf{W}^p)\bar{\mathbf{v}}_k^l - (\bar{\mathbf{v}}_k^l)^T(\mathbf{D}^p+\mathbf{W}^a)\bar{\mathbf{v}}_k^l\right\},\end{aligned} \tag{20}$$

where $\mathbf{v}_j^l$ and $\bar{\mathbf{v}}_k^l$ represent the j-th column vector and k-th row vector of $\mathbf{V}^l$, respectively. Each summand in Eq. (19) and (20) is a quadratic function of a vector variable. Therefore, we can provide upper bounds for these summands:

$$(\mathbf{v}_j^l)^T\mathbf{P}\mathbf{v}_j^l \le \sum_{k=1}^{K}\frac{(\mathbf{P}\mathbf{v}_j^{l,t})_k}{V_{kj}^{l,t}}(V_{kj}^l)^2,$$

$$(\bar{\mathbf{v}}_k^l)^T(\mathbf{D}^a+\mathbf{W}^p)\bar{\mathbf{v}}_k^l \le \sum_{j=1}^{N^l}\frac{((\mathbf{D}^a+\mathbf{W}^p)\bar{\mathbf{v}}_k^{l,t})_j}{V_{kj}^{l,t}}(V_{kj}^l)^2,$$

$$-(\bar{\mathbf{v}}_k^l)^T(\mathbf{D}^p+\mathbf{W}^a)\bar{\mathbf{v}}_k^l \quad \le -\sum_{i,j}(\mathbf{D}^p+\mathbf{W}^a)_{ij}V_{ki}^{l,t}V_{kj}^{l,t}\left(1+\log\frac{V_{ki}^lV_{kj}^l}{V_{ki}^{l,t}V_{kj}^{l,t}}\right),$$

where we use $\mathbf{V}^{l,t}$ to denote the value of $\mathbf{V}^l$ in the t-th iteration of the update algorithm and $\mathbf{v}_j^{l,t}$, $\bar{\mathbf{v}}_k^{l,t}$ are its j-th column vector and k-th row vector, respectively. Note that V_{kj}^l can be viewed both as the k-th element of $\mathbf{v}_j^l$ and as the j-th element of $\bar{\mathbf{v}}_k^l$. The proofs of these bounds follow directly from Lemmas 1 and 2 in [19]. Aggregating the bounds for all the summands, we obtain the auxiliary function for $\mathcal{O}^l(\mathbf{V}^l)$

$$\begin{aligned}&\mathcal{G}^l(\mathbf{V}^{l,t};\mathbf{V}^l)\\=&\frac{1}{2}\sum_{j=1}^{N^l}\sum_{k=1}^{K}\frac{(\mathbf{P}\mathbf{v}_j^{l,t})_k+\beta((\mathbf{D}^a+\mathbf{W}^p)\bar{\mathbf{v}}_k^{l,t})_j}{V_{kj}^{l,t}}(V_{kj}^l)^2\\&-\frac{\beta}{2}\sum_{k=1}^{K}\sum_{i,j}(\mathbf{D}^p+\mathbf{W}^a)_{ij}V_{ki}^{l,t}V_{kj}^{l,t}\left(1+\log\frac{V_{ki}^lV_{kj}^l}{V_{ki}^{l,t}V_{kj}^{l,t}}\right)\\&-\sum_{j=1}^{N^l}\sum_{k=1}^{K}Q_{kj}^lV_{kj}^l.\end{aligned} \tag{21}$$

The estimate of $\mathbf{V}^l$ in the $(t+1)$-th iteration is then computed as

$$\mathbf{V}^{l,t+1} = \arg\min_{\mathbf{V}^l}\mathcal{G}^l(\mathbf{V}^{l,t};\mathbf{V}^l). \tag{22}$$

Differentiating $\mathcal{G}^l(\mathbf{V}^{l,t};\mathbf{V}^l)$ with respect to each V_{kj}^l, we have

$$\begin{aligned}&\frac{\partial \mathcal{G}^l(\mathbf{V}^{l,t};\mathbf{V}^l)}{\partial V_{kj}^l}\\ =&\frac{(\mathbf{P}\mathbf{v}_j^{l,t})_k+\beta((\mathbf{D}^a+\mathbf{W}^p)\bar{\mathbf{v}}_k^{l,t})_j}{V_{kj}^{l,t}}V_{kj}^l-\frac{\beta((\mathbf{D}^p+\mathbf{W}^a)\bar{\mathbf{v}}_k^{l,t})_j}{V_{kj}^l}V_{kj}^{l,t}-Q_{kj}^l\end{aligned}$$

Setting $\partial\mathcal{G}^l(\mathbf{V}^{l,t};\mathbf{V}^l)/\partial V_{kj}^l=0$, we get the update rule for $\mathbf{V}^l$

$$V_{kj}^{l,t+1}=\min\left\{1,V_{kj}^{l,t}\frac{-B_{kj}+\sqrt{B_{kj}^2+4A_{kj}C_{kj}}}{2A_{kj}}\right\}, \tag{23}$$

$$\begin{aligned}A_{kj}=&(\mathbf{P}\mathbf{v}_j^{l,t})_k+\beta((\mathbf{D}^a+\mathbf{W}^p)\bar{\mathbf{v}}_k^{l,t})_j,\\ B_{kj}=&-Q_{kj}^l,C_{kj}=\beta((\mathbf{D}^p+\mathbf{W}^a)\bar{\mathbf{v}}_k^{l,t})_j.\end{aligned}$$

Here $\mathbf{v}_j^l$ and $\bar{\mathbf{v}}_k^l$ denote the j-th column vector and the k-th row vector of $\mathbf{V}^l$, respectively. It is easy to verify that $\mathcal{O}^l(\mathbf{V}^{l,t+1})\leq\mathcal{G}^l(\mathbf{V}^{l,t};\mathbf{V}^{l,t+1})\leq\mathcal{G}^l(\mathbf{V}^{l,t};\mathbf{V}^{l,t})=\mathcal{O}^l(\mathbf{V}^{l,t})$. Therefore, the update rule for $\mathbf{V}^l$ monotonically decreases Eq. 13. The case for $\mathbf{V}^u$ is simpler since we do not have the graph embedding terms:

$$\mathcal{O}^u(\mathbf{V}^u)=\frac{1}{2}tr[(\mathbf{V}^u)^T\mathbf{P}\mathbf{V}^u]-tr[(\mathbf{V}^u)^T\mathbf{Q}^u] \tag{24}$$

Similarly, the auxiliary function for $\mathcal{O}^u(\mathbf{V}^u)$ can be derived

$$\mathcal{G}^u(\mathbf{V}^{u,t};\mathbf{V}^u)=\frac{1}{2}\sum_{j=1}^{N^u}\sum_{k=1}^{K}\frac{(\mathbf{P}\mathbf{v}_j^{u,t})_k}{V_{kj}^{u,t}}(V_{kj}^u)^2-\sum_{j=1}^{N^u}\sum_{k=1}^{K}Q_{kj}^uV_{kj}^u \tag{25}$$

and the update rule can be obtained by setting the partial derivatives to 0:

$$V_{kj}^{u,t+1}=\min\left\{1,V_{kj}^{u,t}\frac{Q_{kj}^u-|Q_{kj}^u|}{2(\mathbf{P}\mathbf{v}_j^{u,t})_k}\right\} \tag{26}$$

5 Experiment

In this section, we conduct the experiments on two real-world data sets to validate the effectiveness of the proposed algorithm MvSL.

Algorithm 1. Optimization of MvSL

```
Data: $\{\mathbf{X}^{(v)}\}_{v=1}^{H}, \alpha, \beta$
Result: $\{\mathbf{U}^{(v)}\}_{v=1}^{H}, \mathbf{V}$
begin
    Randomly initialize $U_{ik}^{(v)} \geq 0, 1 \geq V_{kj} \geq 0, \forall i, j, k, v$
    repeat
        Fix $\mathbf{V}$, update $\mathbf{U}^{(1)}, ..., \mathbf{U}^{(H)}$ as in [12]
        Fix $\mathbf{U}^{(1)}, ..., \mathbf{U}^{(H)}$, update $\mathbf{V}^{l}$ as in (23) and update $\mathbf{V}^{u}$ as in (26) ;
    until convergence or max no. iterations reached
end
```

Table 1. Statistics of the datasets.

Dataset	Size	# of categories	Dimensionality of views
Reuters	1800	6	$21,531/15,506/11,547$
MM2.0	5000	25	$64/144/75/128$

5.1 Data Set

We use two real-world datasets to evaluate the proposed factorization method. The first dataset was constructed from the Reuters Multilingual collection [1]. This test collection contains totally 111,740 Reuters news documents written in five different languages. Documents for each language can be divided into a common set of six categories. Each document was translated into the other four languages and represented as TF-IDF vectors. We took documents written in English as the first view and their Italian and Spanish translations as the second and third views. We randomly sampled 1800 English documents, with 300 for each category. The second dataset came from Microsoft Research Asia Internet Multimedia Dataset 2.0 (MSRA-MM 2.0) [14]. MSRA-MM 2.0 consists of about 1 million images which were respective search results for 1165 popular query concepts in Microsoft Live Search. Each concept has approximately 500-1000 images. For each image, its relevance to the corresponding concept was manually labeled with three levels: very relevant, relevant and irrelevant. 7 different low level features were extracted for each image. To form the experimental dataset, we selected 25 query concepts from the Animal, Object and Scene branches, and then randomly sampled 200 images from each concept while discarding irrelevant ones. We took 4 features in MSRA-MM 2.0 as 4 different views: 64D HSV color histogram, 144D color correlogram, 75D edge distribution histogram and 128D wavelet texture. Hereafter, we refer to the two datasets as Reuters and MM2.0, respectively. The statistics of these datasets are summarized in Table 1.

5.2 Baselines

To validate the performance of our method, we compare the proposed MvSL with the following baselines:

- **NMF** [13].
- Feature concatenation (**ConcatNMF**): This method concatenates feature vectors of different views to form a united representation and then applies NMF.
- Multi-view NMF (**MultiNMF**): MultiNMF [18] is an unsupervised multi-view NMF algorithm.
- Semi-supervised Unified Latent Factor method (**SULF**): SULF [10] is a semi-supervised multi-view nonnegative factorization method which models partial label information as a factorization constraint on $\mathbf{V}^l$.
- Graph regularized NMF (**GNMF**): GNMF [2] is a manifold regularized version of NMF. We extended it to the multi-view case and replaced the affinity graph for approximating data manifolds with the within-class affinity graph defined in Eq. (4) to make it a semi-supervised method on multi-view data.

5.3 Evaluation Metric

Accuracy (ACC) is a typical evaluation metric of classification. Let N_u denote the total number of test images to be labeled, the N_r is the number of images that are assigned the right categories or tags by the proposed algorithms according to the ground truth, the ACC is defined as $\mathrm{ACC=N_r/N_u}$.

Table 2. Classification performance of different factorization methods on the Reuters dataset (accuracy±std dev,%).

Labeled Percentage	NMF-b	ConcatNMF	MultiNMF	SULF	GNMF	MvSL
10	61.55±1.08	63.04±1.67	63.69±1.52	67.93±1.92	68.93±1.77	**70.56±1.21**
20	65.71±1.37	66.09±1.08	67.42±1.97	68.40±1.64	70.59±1.65	**72.67±1.02**
30	67.30±0.27	68.40±1.91	69.16±1.52	70.05±1.48	71.80±1.24	**74.78±1.34**
40	68.41±1.96	69.81±1.96	70.28±1.83	71.86±1.38	72.23±1.54	**75.87±1.26**
50	70.44±1.72	70.75±2.03	71.81±1.47	72.78±1.44	73.78±1.75	**77.33±0.79**

Table 3. Classification performance of different factorization methods on the MM2.0 dataset (accuracy±std dev, %).

Labeled Percentage	NMF-b	ConcatNMF	MultiNMF	SULF	GNMF	MvSL
10	24.56±0.98	27.41±0.83	26.26±0.95	27.47±1.03	28.03±1.17	**30.92±0.44**
20	25.37±0.85	31.24±0.93	30.39±1.12	30.94±1.25	31.55±1.14	**33.83±1.52**
30	26.09±0.71	32.47±0.80	31.85±0.87	33.13±0.87	34.15±0.51	**35.80±0.68**
40	28.03±0.46	34.25±0.71	33.48±0.65	34.94±0.65	35.26±0.97	**37.12±0.73**
50	28.06±0.28	35.08±0.48	34.33±0.56	36.32±0.56	36.61±0.57	**38.16±0.65**

5.4 Experiment Results

Table 2 and Table 3 show the classification performance of different factorization methods on MM2.0 and Reuters, respectively. We varied the percentage of training items from 10% to 50%. The observations are revealed as follows. Firstly, Semi-supervised algorithms are superior to unsupervised algorithms in general, which indicated that exploiting label information could lead to latent spaces with better discriminative structures. Secondly, from comparison between multi-view algorithms and single-view algorithm (NMF), it is easy to see that multi-view algorithms are more preferable for multi-view data. This is in accord with the results of previous multi-view learning work. Thirdly, MvSL and GNMF show superior performance over SULF. SULF models partial label information as a factorization constraint on $\mathbf{V}^l$, which can be viewed as indirect affinity constraints on encoding of within-class items. On the contrary, the graph embedding terms in MvSL and GNMF impose direct affinity constraints on item encodings and therefore could lead to more explicit semantic structures in the learned latent spaces. Finally, MvSL outperformed the baseline methods under all cases. The reason should be that MvSL not only directly exploits label information via a graph embedding framework, but also adds regularization by $L_{1,2}$-norm on $\mathbf{U}^{(v)}$ successfully promotes that sparsity pattern is shared among data items or features within classes. These properties could help to learn a clearer semantic latent space.

5.5 Parameter Sensitive Analysis

There are two essential parameters in new methods. β measures the importance of the semi-supervised part of MvSL (i.e. the graph embedding regularization terms), while α controls the degree of sparsity of the basis matrices. We investigate their influence on MvSL's performance by varying one while fixing the other one.

The classification results are shown in Figure 1 for MM2.0 and Reuters. We found the general behavior of the two parameters was the same: when increasing the parameter from 0, the performance curves first went up and then went down. This indicates that when assigned moderate weights, the sparseness and semi-supervised constraints indeed helped learn a better latent subspace. Based observations , we set $\alpha = 10$, $\beta = 0.02$ for experiments.

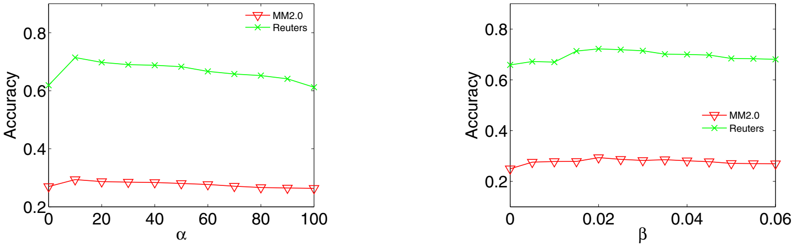

Fig. 1. Influence of different parameter settings on the performance of MvSL: (a) varying α while setting $\beta = 0.02$, (b) varying β while setting $\alpha = 10$

6 Conclusion

We have proposed Multi-view semantic learning (MvSL), a novel nonnegative latent representation learning algorithm for representation learning multi-view data. MvSL tries to learn a semantic latent subspace of items by exploiting both multiple views of items and partial label information. The partial label information was used to construct a graph embedding framework, which encouraged items of the same category to be near with each other and kept items belonging to different categories as distant as possible in the latent subspace. What's more, kernel alignment effectively estimated the items pair similarity among multi-view data, which further extended graph embedding framework. Another novel property of MvSL was that it allowed each latent dimension to be associated with a subset of views by imposing $L_{1,2}$-norm on each basis $\mathbf{U}^{(v)}$. Therefore, MvSL is able to learn flexible latent factor sharing structures which could lead to more meaningful semantic latent subspaces. An efficient multiplicative-based iterative algorithm is developed to solve the proposed optimization problem. The classification experimental results on two real-world data sets have demonstrated the effectiveness of our method.

Graph embedding is a general framework and different definitions of the within class affinity graph G^a and the discriminative graph G^p can be employed. How to propose more suitable similarity criteria with multi-view data is an interesting direction for further study.

Acknowledgments. This research was supported by the National High-tech R&D Program of China (863 Program) (No. 2014AA015201), National Natural Science Foundation of China (No. 61373118).

References

1. Amini, M., Usunier, N., Goutte, C.: Learning from multiple partially observed views-an application to multilingual text categorization. In: Advances in Neural Information Processing Systems, pp. 28–36 (2009)
2. Cai, D., He, X., Han, J., Huang, T.S.: Graph regularized nonnegative matrix factorization for data representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(8), 1548–1560 (2011)
3. Chen, N., Zhu, J., Xing, E.P.: Predictive subspace learning for multi-view data: a large margin approach. In: Advances in Neural Information Processing Systems, pp. 361–369 (2010)
4. Han, Y., Wu, F., Tao, D., Shao, J., Zhuang, Y., Jiang, J.: Sparse unsupervised dimensionality reduction for multiple view data. IEEE Transactions on Circuits and Systems for Video Technology **22**(10), 1485–1496 (2012)
5. He, J., Chang, S.-F., Xie, L.: Fast kernel learning for spatial pyramid matching. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–7. IEEE (2008)
6. Hidru, D., Goldenberg, A.: Equinmf: Graph regularized multiview nonnegative matrix factorization (2014). arXiv preprint arXiv:1409.4018
7. Hotelling, H.: Relations between two sets of variates. Biometrika 321–377 (1936)

8. Hoyer, P.O.: Non-negative sparse coding. In: Proceedings of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing, pp. 557–565 (2002)
9. Jia, Y., Salzmann, M., Darrell, T.: Factorized latent spaces with structured sparsity. In: Advances in Neural Information Processing Systems, pp. 982–990 (2010)
10. Jiang, Y., Liu, J., Li, Z., Lu, H.: Semi-supervised unified latent factor learning with multi-view data. Machine Vision and Applications **25**(7), 1635–1645 (2014)
11. Kalayeh, M., Idrees, H., Shah, M.: NMF-KNN: Image annotation using weighted multi-view non-negative matrix factorization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 184–191 (2014)
12. Kim, J., Monteiro, R., Park, H.: Group sparsity in nonnegative matrix factorization. In: SDM, pp. 851–862. SIAM (2012)
13. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)
14. Li, H., Wang, M., Hua, X.-S.: Msra-mm 2.0: A large-scale web multimedia dataset. In: IEEE International Conference on Data Mining Workshops, pp. 164–169. IEEE (2009)
15. Lin, C.-J.: Projected gradient methods for nonnegative matrix factorization. Neural computation **19**(10), 2756–2779 (2007)
16. Liu, H., Wu, Z., Li, X., Cai, D., Huang, T.S.: Constrained nonnegative matrix factorization for image representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(7), 1299–1311 (2012)
17. Liu, J., Jiang, Y., Li, Z., Zhou, Z.-H., Lu, H.: Partially shared latent factor learning with multiview data (2014)
18. Liu, J., Wang, C., Gao, J., Han, J.: Multi-view clustering via joint nonnegative matrix factorization. In: Proc. of SDM, vol. 13, pp. 252–260 (2013)
19. Sha, F., Lin, Y., Saul, L.K., Lee, D.D.: Multiplicative updates for nonnegative quadratic programming. Neural Computation **19**(8), 2004–2031 (2007)
20. Shawe-Taylor, N., Kandola, A.: On kernel target alignment. Advances in neural information processing systems **14**, 367 (2002)
21. Wang, Y., Jia, Y.: Fisher non-negative matrix factorization for learning local features. In: Proc. Asian Conf. on Comp. Vision. Citeseer (2004)
22. Xia, T., Tao, D., Mei, T., Zhang, Y.: Multiview spectral embedding. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics **40**(6), 1438–1446 (2010)
23. Xu, C., Tao, D., Xu, C.: A survey on multi-view learning. arXiv preprint arXiv:1304.5634 (2013)
24. Yan, S., Xu, D., Zhang, B., Zhang, H.-J., Yang, Q., Lin, S.: Graph embedding and extensions: a general framework for dimensionality reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(1), 40–51 (2007)
25. Zafeiriou, S., Tefas, A., Buciu, I., Pitas, I.: Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification. IEEE Transactions on Neural Networks **17**(3), 683–695 (2006)