Multi-View Concept Learning for Data Representation

Ziyu Guan, Lijun Zhang, Jinye Peng, and Jianping Fan

Abstract—Real-world datasets often involve multiple views of data items, e.g., a Web page can be described by both its content and anchor texts of hyperlinks leading to it; photos in Flickr could be characterized by visual features, as well as user contributed tags. Different views provide information complementary to each other. Synthesizing multi-view features can lead to a comprehensive description of the data items, which could benefit many data analytic applications. Unfortunately, the simple idea of concatenating different feature vectors ignores statistical properties of each view and usually incurs the "curse of dimensionality" problem. We propose Multi-view Concept Learning (MCL), a novel nonnegative latent representation learning algorithm for capturing conceptual factors from multi-view data. MCL exploits both multi-view information and label information. The key idea is to learn a common latent space across different views which (1) captures the semantic relationships between data items through graph embedding regularization on labeled items, and (2) allows each latent factor to be associated with a subset of views via sparseness constraints. In this way, MCL could capture flexible conceptual patterns hidden in multi-view features. Experiments on a toy problem and three real-world datasets show that MCL performs well and outperforms baseline methods.

Index Terms—Multi-view learning, nonnegative matrix factorization, graph embedding, structured sparsity

1 Introduction

Many real-world data analytic problems involve rich data which consists of multiple modalities or views of data items. For example, the same news story can be written in different languages; in recommender systems a user's potential interests are reflected by not only her blog articles but also her social circles; in a photo sharing Website (e.g. Flickr), images could be indexed with various visual features as well as tags contributed by users. Generally speaking, different views represent features of the data items from different perspectives and therefore often provide information complementary to each other. A good synthesization of multi-view features can lead to a more comprehensive description of the data items, which could benefit many related tasks such as classification, clustering and retrieval. For instance, by considering both user generated content and social relationships, the recommendation performance can be boosted [1]; by combining multiple (visual) features of images, we could also improve the performance of image classification [2], [3], search [4], etc.

A straightforward solution for synthesizing multi-view features is to concatenate the feature vectors to form a new vector and feed the new vector into machine learning algorithms. However, this method ignores the specific statistical properties of different views and usually incurs the *curse of dimensionality* problem [5]. In recent years, multi-view learning has attracted more and more attention from the research community. Multi-view learning approaches have been proposed in various contexts, e.g. co-training of classifiers [6], multi-view clustering [7], [8], multiple kernel learning [9], [10], multi-view transfer learning [11], [12], etc. A growing area in the multi-view learning literature is *multi-view latent subspace learning*. The goal is to obtain a unified latent subspace (i.e., representation) shared by multiple views. The dimensionality of the subspace is typically lower than that of any single view, consequently alleviating the curse of dimensionality problem. The earliest method, Canonical Correlation Analysis (CCA) [13], tries to extract a common subspace for two views by maximizing the correlations between their projections onto the subspace. Recently, a lot of techniques have been generalized or applied to multi-view subspace learning, such as matrix factorization [2], [14], [15], [16], [17], spectral embedding [18], undirected graphical models [19] and Gaussian processes [20].

Matrix factorization is a promising technique for latent representation learning, since the learned latent factors could be interpreted by the corresponding basis components and we can conveniently regularize the factorized matrices for different purposes. Among the matrix factorization methods, Nonnegative Matrix Factorization (NMF) [21] is a popular one. In NMF, Each data item is reconstructed as an additive linear combination of nonnegative basis components. Therefore, NMF has the intuitive notion of combining parts to form the whole, which conforms to the cognitive process of human brains from psychological and physiological evidences. Recently, researchers have also proposed variants of NMF for multi-view problems [3], [16], [17].

In this work, we are concerned with nonnegative latent representation learning from multi-view features. Since the learned latent space is typically used for semantic tasks

- *Z. Guan, J. Peng, and J. Fan are with the College of Information and Technology, Northwest University of China, Xi'an, CN 710127. E-mail: {ziyuguan, pjy, jfan}@nwu.edu.cn.*
- *L. Zhang is with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China. E-mail: zhanglj@lamda.nju.edu.cn.*

Manuscript received 15 Oct. 2014; revised 15 Apr. 2015; accepted 4 June 2015. Date of publication 21 June 2015; date of current version 2 Oct. 2015. Recommended for acceptance by S. Yan.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2015.2448542

(e.g. classification), a good learning algorithm should be able to capture the conceptual structures in multi-view data. A conceptual latent space learned from multi-view data should possess the following properties: (1) The semantic relationships between data items are preserved in the latent space. (2) Each latent factor (i.e., latent dimension) of the latent space has the flexibility of being associated with a subset of views [14]. This is because the conceptual patterns hidden in multi-view data might not be associated with all the views. For example, people follow their own tastes for some kinds of interests (e.g. clothes), while for interests such as "electronic products" they are much more likely to be influenced by friends and follow their recommendations; Images of concept "striped shirts" exhibit visual patterns which are associated with texture features but independent on color features, since the stripes can be in different colors. However, to our knowledge no existing multi-view latent space learning algorithm aimed to find a latent space which possesses both properties described above.

We propose in this paper a novel nonnegative latent representation learning algorithm, namely, Multi-view Concept Learning (MCL), for multi-view data. MCL jointly factorizes data matrices (containing feature vectors of data items as columns) of different views. Each data matrix is factorized into a *basis matrix* and an *encoding matrix* where the encoding matrix is shared among different views and is called the *consensus encoding matrix*. The intuition behind this joint factorization scheme is that each dimension of the consensus latent space represents a conceptual pattern which is interpreted by the combination of corresponding basis vectors from different views. We regularize this basic factorization framework to encourage the aforementioned two properties. First, in order to ensure the learned latent space captures the semantic relationships between data items, partial label information is incorporated into the factorization framework via imposing graph embedding [22] regularization on the consensus encodings directly. The general idea is that we encourage items of the same category to be close to each other while keep those belonging to different categories far away from each other in the learned latent space. Second, we add structured sparseness constraints on the basis matrices to allow a latent dimension to be associated with a subset of views. Specifically, for each view's basis matrix we add a $L_{1,\infty}$ norm regularizer to encourage some basis vectors to be zeroed-out [14]. By imposing such structured sparseness constraints, some latent dimensions would be explained by subsets of views rather than by all the views only. Finally, an L_1 regularizer is imposed on the consensus encoding matrix since a data item usually does not possess many conceptual features. This would force the latent space to represent information compactly.

We develop an optimization method for MCL which optimizes the objective function with respect to the basis matrices and the consensus encoding matrix alternately. For the sub-problems involving basis matrices, we develop an efficient optimization algorithm based on the composite gradient mapping method which has been proved to converge very fast [23]. For the sub-problem of the consensus encoding matrix, a multiplicative update algorithm is proposed. We also analyze the computational complexity of the proposed optimization method. For empirical evaluation, a toy factorization problem and three real-world datasets are employed. Experimental results on these datasets indicate that MCL is effective and outperforms baselines significantly.

The contributions of this work are summarized as follows: (1) We propose the problem of learning a nonnegative conceptual representation from multi-view data. In order to capture the flexible conceptual patterns in multi-view data, the learned latent space ought to not only preserve the semantic relationships between data items, but also allow each latent factor to be associated with a subset of views. Nevertheless, no existing work targeted the same goal. (2) We propose a novel nonnegative latent representation learning algorithm MCL to address this problem. MCL forces the learned latent space to possess the aforementioned two properties by graph embedding regularization on labeled items' encodings and structured sparseness regularization on basis matrices. (3) We develop a block coordinate descent method to solve the optimization problem of MCL. The optimization of basis matrices is based on the recently proposed composite gradient mapping technique [23]. For the consensus encoding matrix, we develop a multiplicative update algorithm. The optimization method is efficient, as we will show in complexity analysis. (4) We evaluate MCL qualitatively on a synthetic toy dataset and quantitatively on three real-world datasets. The results show that MCL works as expected and is significantly superior over baseline methods.

2 A Brief Review of NMF and Related Variants

In this section we briefly review NMF and some follow-up variants which are related to our work. We begin with a description of common notations in this paper.

2.1 Common Notations

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

2.2 NMF and Related Variants

NMF [21] is designed for analyzing data matrices with nonnegative elements. Given data matrix $\mathbf{X}$ and a pre-specified positive integer $K < \min(M, N)$, the goal is to find two nonnegative matrices $\mathbf{U} \in \mathbb{R}_+^{M \times K}$ and $\mathbf{V} \in \mathbb{R}_+^{K \times N}$ such that their product well approximates the original data matrix:

$$\mathbf{X} \approx \mathbf{U}\mathbf{V}.$$

When each column of $\mathbf{X}$ represents an item, NMF can be interpreted as approximating it by a linear combination of k "basis" columns in $\mathbf{U}$ where the combination weights come from the corresponding column of $\mathbf{V}$. Therefore, the basis matrix $\mathbf{U}$ and the encoding matrix $\mathbf{V}$ together define a latent space representation for the data items. Due to the nonnegative constraints, the factorized latent space has the intuitive notion of combining parts to form an object [21].

In order to learn $\mathbf{U}$ and $\mathbf{V}$, Lee and Seung introduced two cost functions to quantify the quality of the approximation: a least square cost and a divergence-style cost [21]. In this paper, we are focused on the least square cost which is defined as

$$\mathcal{O} = \sum_{i=1}^{M}\sum_{j=1}^{N}\left(X_{ij} - \sum_{k=1}^{K} U_{ik}V_{kj}\right)^2 = \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2. \tag{1}$$

The optimization problem in terms of $\mathcal{O}$ can be formulated as

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

The objective function $\mathcal{O}$ is not convex when both $\mathbf{U}$ and $\mathbf{V}$ are taken as variables. However, $\mathcal{O}$ is convex in $\mathbf{U}$ when $\mathbf{V}$ is fixed and vice versa. Lee and Seung [21] presented an iterative multiplicative update algorithm to find a local minimum of $\mathcal{O}$. In the tth iteration, new values of $\mathbf{U}$ and $\mathbf{V}$ are calculated from their values in the last iteration:

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

By constructing auxiliary functions, it is proved that $\mathcal{O}$ is non-increasing under the above update rules.

The basic NMF model has been extended in various ways in recent years. We just name a few which are related to our work. A comprehensive survey can be found in [24]. One direction related to our work is incorporating label information into NMF [25], [26]. These works added discriminative information into NMF via regularizing the encodings of items (i.e., columns of $\mathbf{V}$) by fisher-style discriminative constraints. Our method also exploits label information through regularizing the encoding matrix $\mathbf{V}$. Nevertheless, our regularization is defined as a general graph embedding framework, with fisher discriminative analysis as its special case [22]. Another related variant of NMF is sparse NMF [27], [28], [29]. Sparseness constraints not only encourage local and compact representations, but also improve the stability of the factorization. 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 $\mathbf{V}$. In our case, we also impose a L_1 penalty on $\mathbf{V}$. However, the story for the basis part 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, structured sparseness penalties [14] are used to achieve this goal.

Recently, NMF has also been adapted to the multi-view setting in specific contexts, e.g. clustering [17], image annotation [16] and semi-supervised learning [3]. However, none of these works aimed to learn latent spaces which preserve semantic relationships between items and meanwhile, allow latent dimensions to be associated with subsets of views. In [3], Jiang et al. proposed a semi-supervised Multi-view NMF which also exploited partial label information of data items. Our MCL is different from their method from two aspects: (1) In their method label information was incorporated as a factorization constraint on $\mathbf{V}$, i.e., reconstructing the label indicator matrix through multiplying $\mathbf{V}$ by a weight matrix. Hence, their method intrinsically imposed *indirect* affinity constraints on encodings of labeled items, while MCL *directly* penalizes the distances between labeled items in the latent space. (2) their model did not have sparseness constraints, while our model employs sparseness constraints to learn flexible latent factors. We will also compare MCL with the method in [3] in experiments.

3 Multi-View Concept Learning

In this section, we introduce Multi-view Concept Learning. Fig. 1 illustrates the work flow of MCL. From the set of items, we first obtain various features to construct the set of data matrices $\{\mathbf{X}^{(v)}\}_{v=1}^{H}$ where $\mathbf{X}^{(v)} \in \mathbb{R}_{+}^{M_v \times N}$. Then the labeled items are used to construct the within-class affinity graph G^a and the between-class penalty graph G^p. Generally speaking, G^a encourages items with the same label to be close to one another in the learned latent space, while G^p tries to keep items of different classes far away from each other. The data matrices are then factorized into basis matrices ($\{\mathbf{U}^{(v)}\}_{v=1}^{H}$) and the consensus encoding matrix $\mathbf{V}$. Note that in Fig. 1 fully white elements in the matrices mean their values are 0. By imposing a structured sparseness constraint on each $\mathbf{U}^{(v)}$, some basis vectors could be zeroed-out so that the corresponding latent dimensions do not depend on that view. For example, in Fig. 1 the second column of $\mathbf{U}^{(1)}$ and the third column of $\mathbf{U}^{(2)}$ are zero columns, which means the second latent dimension is not associated with view 1 and the third is not associated with view 2. For $\mathbf{V}$, we add a sparseness constraint in addition to the graph embedding regularization to prevent an item from possessing too many latent features. In the following, we discuss the design choices and formulate the optimization problem of MCL.

3.1 Multi-View NMF

The consensus principle is the fundamental principle in multi-view learning [30]. The basic optimization framework of MCL tries to learn a common latent space via a shared encoding matrix $\mathbf{V}$ [14], [16], [17]:

$$\begin{aligned}&\min_{\{\mathbf{U}^{(v)}\}_{v=1}^{H},\mathbf{V}} \ \frac{1}{2}\sum_{v=1}^{H}\|\mathbf{X}^{(v)} - \mathbf{U}^{(v)}\mathbf{V}\|_F^2\\ &\text{s.t.} \quad U_{ik}^{(v)} \geq 0, V_{kj} \geq 0, \ \ \forall i,j,k,v.\end{aligned} \tag{3}$$

In this way, each item is forced to have the same encoding under different views and the basis matrices of different views are coupled together through $\mathbf{V}$. However, such an unsupervised framework cannot guarantee that the learned latent space captures the conceptual structures in multi-view data. Next, we introduce the semi-supervised part of the model.

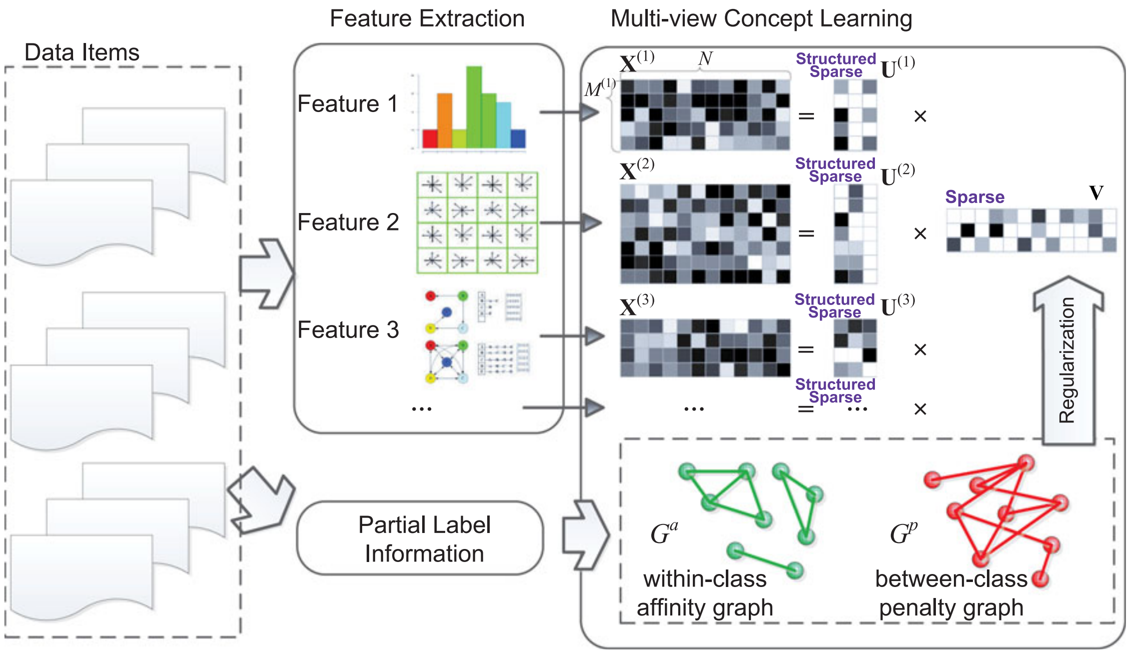


Fig. 1. An illustration of the work flow of the proposed approach. Fully White color in the matrices means value 0.

3.2 Semi-Supervision on $\mathbf{V}$

In order to let the learned latent space reflect the semantic relationships between items, we propose to regularize the consensus encoding matrix $\mathbf{V}$ by a graph embedding framework [22]. Since we assume only partial label information is obtained, $\mathbf{V}$ and each data matrix can be divided into labeled part and unlabeled part. We assume that columns in each $\mathbf{X}^{(v)}$ are arranged such that labeled items come before unlabeled ones. Hence, $\mathbf{X}^{(v)}$ and $\mathbf{V}$ can be written as $\mathbf{X}^{(v)} = [\mathbf{X}^{(v),l}\ \mathbf{X}^{(v),u}]$ and $\mathbf{V} = [\mathbf{V}^l\ \mathbf{V}^u]$, where the superscripts l and u mean "labeled" and "unlabeled", respectively. As shown in Fig. 1, we construct two graphs, the within-class affinity graph G^a and the between-class penalty graph G^p, connecting within-class items and between-class items, respectively. The nodes in the graphs are labeled items and the edges are specified by the corresponding weighted adjacency matrices, denoted by $\mathbf{W}^a$ and $\mathbf{W}^p$. Let $\mathbf{v}_i^l$ be the ith column of $\mathbf{V}^l$. The graph embedding objectives are defined as follows:

$$\min_{\mathbf{V}^l} \frac{1}{2}\sum_{i=1}^{N^l}\sum_{j=1}^{N^l} W_{ij}^a \|\mathbf{v}_i^l - \mathbf{v}_j^l\|_2^2 = \min_{\mathbf{V}^l} \frac{1}{2} tr[\mathbf{V}^l \mathbf{L}^a (\mathbf{V}^l)^T], \quad (4)$$

$$\max_{\mathbf{V}^l} \frac{1}{2}\sum_{i=1}^{N^l}\sum_{j=1}^{N^l} W_{ij}^p \|\mathbf{v}_i^l - \mathbf{v}_j^l\|_2^2 = \max_{\mathbf{V}^l} \frac{1}{2} tr[\mathbf{V}^l \mathbf{L}^p (\mathbf{V}^l)^T], \quad (5)$$

where $tr(\cdot)$ denotes the trace of a matrix, N^l is the number of labeled items and $\mathbf{L}^a = \mathbf{D}^a - \mathbf{W}^a$ is the graph Laplacian matrix for G^a with the (i,i)th element 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. (4) means items belonging to the same class should be near each other in the learned latent space, while Eq. (5) tries to keep items from different classes as distant as possible. However, only with the nonnegative constraints Eq. (5) would diverge. Note that there is an arbitrary scaling factor in solutions to problem (3): for any invertible $K \times K$ matrix $\mathbf{Q}$, we have $\mathbf{U}^{(v)}\mathbf{V} = (\mathbf{U}^{(v)}\mathbf{Q})(\mathbf{Q}^{-1}\mathbf{V})$. It means for any solution $< \{\mathbf{U}^{(v)}\}_{v=1}^H, \mathbf{V} >$ of (3), we can always find a proper $\mathbf{Q}$ such that $< \{\mathbf{U}^{(v)}\mathbf{Q}\}_{v=1}^H, \mathbf{Q}^{-1}\mathbf{V} >$ is an equivalent solution and all elements of $\mathbf{Q}^{-1}\mathbf{V}$ are within $[0,1]$. Therefore, without loss of generality, we add the constraints $\{V_{kj} \le 1, \forall k, j\}$ to put an upper bound on (5).

The above graph embedding regularization framework is general and can be instantiated by a specification of $\mathbf{W}^a$ and $\mathbf{W}^p$. In this work, we define $\mathbf{W}^a$ and $\mathbf{W}^p$ as

$$W_{ij}^a = \begin{cases} \frac{1}{N_{c_i}^l} - \frac{1}{N^l}, & \text{if } c_i = c_j \\ 0, & \text{otherwise} \end{cases}, \quad (6)$$

$$W_{ij}^p = \begin{cases} \frac{1}{N^l}, & \text{if } c_i \neq c_j \\ 0, & \text{otherwise} \end{cases}, \quad (7)$$

where c_i denotes the label of item i and $N_{c_i}^l$ is the total number of items with label c_i. Although more sophisticated definitions for $\mathbf{W}^a$ and $\mathbf{W}^p$ (such as the maximum margin style definitions in [22]) are available, they require additional computation for similarity estimation and nearest neighbor search. In this work we employ the above definitions for simplicity and efficiency.

3.3 Sparseness Constraints

For each $\mathbf{U}^{(v)}$, a structured sparseness regularizer is added to the objective function to encourage some basis column vectors in $\mathbf{U}^{(v)}$ to become 0. This makes view v independent of the latent dimensions which correspond to these zero-valued basis vectors. Let $\mathbf{U}$ be the basis matrix for an arbitrary view. One can achieve structured sparsity via $L_{1,q}$ norm where q is an integer ranging from 1 to ∞:

$$\|\mathbf{U}\|_{1,q} = \sum_{k=1}^{K} \|\mathbf{u}_k\|_q,$$

where $\mathbf{u}_k$ represents the kth column of $\mathbf{U}$. To keep the optimization problem convex in $\mathbf{U}$, $q = 2$ and $q = \infty$ are common choices. In this work, we choose $p = \infty$ since it has been shown that $L_{1,\infty}$ is more effective than $L_{1,2}$ [31]. $L_{1,\infty}$ norm of matrix $\mathbf{U}$ is defined as

$$\|\mathbf{U}\|_{1,\infty} = \sum_{k=1}^{K} \max_{1 \le i \le M} |U_{ik}|. \tag{8}$$

Finally, the L_1 norm of each item's encoding is added to the objective function to make the encoding sparse with the intuition that an item cannot possess too many conceptual features:

$$\|\mathbf{V}\|_{1,1} = \sum_{i=1}^{N} \sum_{k=1}^{K} |V_{ki}|. \tag{9}$$

3.4 Objective Function of MCL

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

$$\begin{aligned} \min_{\{\mathbf{U}^{(v)}\}_{v=1}^{H}, \mathbf{V}} \quad & \frac{1}{2}\sum_{v=1}^{H} \|\mathbf{X}^{(v)} - \mathbf{U}^{(v)}\mathbf{V}\|_F^2 + \alpha \sum_{v=1}^{H} \|\mathbf{U}^{(v)}\|_{1,\infty} \\ & + \frac{\beta}{2}\{tr[\mathbf{V}^l\mathbf{L}^a(\mathbf{V}^l)^T] - tr[\mathbf{V}^l\mathbf{L}^p(\mathbf{V}^l)^T]\} \\ & + \gamma\|\mathbf{V}\|_{1,1} \\ \text{s.t.} \quad & U_{ik}^{(v)} \ge 0, 1 \ge V_{kj} \ge 0, \quad \forall i, j, k, v. \end{aligned} \tag{10}$$

Since we constrain elements of $\mathbf{V}$ to be in $[0, 1]$, this optimization problem is well lower bounded. Note that the scaling issue (i.e., scaling up $\mathbf{U}$ and scaling down $\mathbf{V}$ by the same factor do not affect the least square terms but always decrease the $L_{1,1}$ term) in the nonnegative sparse coding problem [27] does not exist here due to the structured sparseness regularization on $\mathbf{U}$'s. The way we incorporate label information is similar to the previous Fisher-NMF works [25], [32]. However, the graph embedding framework in our model is general and the fisher terms can be taken as a concrete instantiation.

The optimization problem (10) is not convex in both $\{\mathbf{U}^{(v)}\}_{v=1}^{H}$ and $\mathbf{V}$. Therefore, we can only find its local minima. In the next section, we develop an efficient optimization method for (10).

4 Optimization

When $\{\mathbf{U}^{(v)}\}_{v=1}^{H}$ are fixed, (10) is convex in $\mathbf{V}$, and vice versa. Therefore, we propose to solve MCL by a block coordinate descent method [33] which optimizes one block of variables ($\{\mathbf{U}^{(v)}\}_{v=1}^{H}$ or $\mathbf{V}$) while keeping the other block fixed. The procedure is depicted in Algorithm 1. Next, we describe the detailed ideas for addressing the two subproblems (lines 4 and 5 in Algorithm 1).

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

It is easy to see that, given $\mathbf{V}$ the $\mathbf{U}^{(v)}$'s 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 subproblem involving $\mathbf{U}$ can be written as

$$\begin{aligned} \min_{\mathbf{U}} \ & \phi(\mathbf{U}) := \left(\frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2 + \alpha\|\mathbf{U}\|_{1,\infty}\right) \\ \text{s.t.} \ & U_{ik} \ge 0, \ \forall i, k. \end{aligned} \tag{11}$$

Algorithm 1. Optimization of MCL

```
  Input: $\{\mathbf{X}^{(v)}\}_{v=1}^{H}$, $\alpha$, $\beta$, $\gamma$
  Output: $\{\mathbf{U}^{(v)}\}_{v=1}^{H}$, $\mathbf{V}$
1 begin
2   Randomly initialize $U_{ik}^{(v)} \ge 0, 1 \ge V_{kj} \ge 0, \forall i, j, k, v.$ ;
3   repeat
4       Optimize problem (10) with respect to $\{\mathbf{U}^{(v)}\}_{v=1}^{H}$ while keeping $\mathbf{V}$ fixed. ;
5       Optimize problem (10) with respect to $\mathbf{V}$ while keeping $\{\mathbf{U}^{(v)}\}_{v=1}^{H}$ fixed.
6   until convergence or max no. iterations reached;
7 end
```

$\phi(\mathbf{U})$ is a composite objective function where the first term is strongly convex and differentiable, and the second term is convex. We develop an optimization algorithm based on the composite gradient mapping technique proposed for minimizing composite objective functions [23]. The idea is to iteratively minimize an auxiliary function and adjust the guess of the Lipschitz constant of the first term of $\phi(\mathbf{U})$ so that we decrease the objective function as fast as possible. Let $f(\mathbf{U}) = \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2$ and $\mathbf{U}^t$ be the value of $\mathbf{U}$ in the tth iteration. In our case, the auxiliary function is defined as

$$\begin{aligned} m_L(\mathbf{U}^t; \mathbf{U}) = & f(\mathbf{U}^t) + tr[\nabla f(\mathbf{U}^t)^T(\mathbf{U} - \mathbf{U}^t)] \\ & + \frac{L}{2}\|\mathbf{U} - \mathbf{U}^t\|_F^2 + \alpha\|\mathbf{U}\|_{1,\infty}, \end{aligned} \tag{12}$$

where L is the estimate of the Lipschitz constant, L_f, of $f(\cdot)$, and $\nabla f(\mathbf{U}^t)$ is the gradient of $f(\cdot)$ at $\mathbf{U}^t$:

$$\nabla f(\mathbf{U}^t) = \mathbf{U}^t\mathbf{V}\mathbf{V}^T - \mathbf{X}\mathbf{V}^T. \tag{13}$$

By minimizing $m_L(\mathbf{U}^t; \mathbf{U})$ with the nonnegative constraints, we obtain a candidate for $\mathbf{U}^{t+1}$ which is denoted by $T_L(\mathbf{U}^t)$:

$$T_L(\mathbf{U}^t) = \arg\min_{U_{ik} \ge 0, \forall i,k} m_L(\mathbf{U}^t; \mathbf{U}) \tag{14}$$

Note that it is guaranteed that $m_L(\mathbf{U}^t; T_L(\mathbf{U}^t)) \le \phi(\mathbf{U}^t)$ since $m_L(\mathbf{U}^t; \mathbf{U}^t) = \phi(\mathbf{U}^t)$ and $T_L(\mathbf{U}^t)$ is the minimizer of $m_L(\mathbf{U}^t; \mathbf{U})$. It has been proved that for $L \ge L_f$ we have $\phi(T_L(\mathbf{U}^t)) \le m_L(\mathbf{U}^t; T_L(\mathbf{U}^t))$ [23]. Therefore, the optimization algorithm starts with an estimate L_0 such that $0 < L_0 \le L_f$, and in each iteration adjusts L until we get $\phi(T_L(\mathbf{U}^t)) \le m_L(\mathbf{U}^t; T_L(\mathbf{U}^t))$. The algorithm is shown in Algorithm 2. The inner loop (lines 5-11) tries to find an acceptable $\mathbf{U}^{t+1}$. L can keep increasing only if $L \le L_f$. Then L is scaled down by η_d (line 13) since the step size (i.e., $\|T_L(\mathbf{U}^t) - \mathbf{U}^t\|$) is inversely proportional to L. When $\eta_u = \eta_d = 2$, the total number of inner iterations after t outer iterations can be upper bounded by $2(t+1) + \log_2 \frac{L_f}{L_0}$ [23]. Algorithm 2 was proved to converge as $O(1/T)$ where T is the total number of outer iterations [23]. Nesterov also proposed an accelerated version of the algorithm with convergence rate $O(1/T^2)$.

Algorithm 2. Composite Gradient Mapping

```
   Input: $\eta_u > 1, \eta_d > 1$: scaling parameters for $L$
1  begin
2      Initialize $U^0_{ik} \geq 0, \forall i, k$, and $L_0 : 0 < L_0 \leq L_f$. ;
3      $t = 0$ ;
4      repeat
5          repeat
6              $L = L_t$;
7              Optimize (14) to get $T_L(\mathbf{U}^t)$ ;
8              if $\phi(T_L(\mathbf{U}^t)) > m_L(\mathbf{U}^t; T_L(\mathbf{U}^t))$ then
9                  $L = L\eta_u$
10             end
11         until $\phi(T_L(\mathbf{U}^t)) \leq m_L(\mathbf{U}^t; T_L(\mathbf{U}^t))$;
12         $\mathbf{U}^{t+1} = T_L(\mathbf{U}^t)$;
13         $L_{t+1} = \max(L_0, L/\eta_d)$;
14         $t = t + 1$
15     until convergence;
16 end
```

The remaining problem is how to optimize (14) efficiently since we need to employ its solver as a routine in the inner loop of Algorithm 2. We propose a linear time solver for (14). First, we can rewrite $m_L(\mathbf{U}^t; \mathbf{U})$ as follows (let $\mathbf{Y} := \mathbf{U} - \mathbf{U}^t$ for clarity)

$$\begin{aligned} m_L(\mathbf{U}^t; \mathbf{U}) &= \frac{L}{2}\|\mathbf{Y}\|_F^2 + tr[\nabla f(\mathbf{U}^t)^T\mathbf{Y}] + \alpha\|\mathbf{U}\|_{1,\infty} + f(\mathbf{U}^t) \\ &= \frac{L}{2}\left\{\|\mathbf{Y}\|_F^2 + \frac{2}{L}tr[\nabla f(\mathbf{U}^t)^T\mathbf{Y}] + \frac{1}{L^2}\|\nabla f(\mathbf{U}^t)\|_F^2\right\} \\ &\quad + \alpha\|\mathbf{U}\|_{1,\infty} + f(\mathbf{U}^t) - \frac{1}{2L}\|\nabla f(\mathbf{U}^t)\|_F^2 \\ &= \frac{L}{2}\|\mathbf{Y} + \frac{1}{L}\nabla f(\mathbf{U}^t)\|_F^2 + \alpha\|\mathbf{U}\|_{1,\infty} + const. \end{aligned}$$

Substituting $\mathbf{U} - \mathbf{U}^t$ for $\mathbf{Y}$, the optimization problem (14) becomes

$$\begin{aligned} \min_{\mathbf{U}} \quad & \frac{L}{2}\|\mathbf{U} - \mathbf{U}^t + \frac{1}{L}\nabla f(\mathbf{U}^t)\|_F^2 + \alpha\sum_{k=1}^{K}\max_{1\leq i\leq M}|U_{ik}| \\ \text{s.t.} \quad & U_{ik} \geq 0, \quad \forall i, k. \end{aligned} \tag{15}$$

The Frobenius norm term is intrinsically a summation over the costs of all $\mathbf{U}$'s elements. The second term sums over the infinity norms of all columns of $\mathbf{U}$. Hence, (15) can be further decomposed into independent optimization problems for different columns of $\mathbf{U}$. Let $\mathbf{u}$ be an arbitrary column of $\mathbf{U}$ and $\mathbf{b}$ be the column of $\left(\mathbf{U}^t - \frac{1}{L}\nabla f(\mathbf{U}^t)\right)$ at the same index. The problem for this column can be written as

$$\begin{aligned} \min_{\mathbf{u}} \quad & \frac{1}{2}\|\mathbf{u} - \mathbf{b}\|_2^2 + \frac{\alpha}{L}\|\mathbf{u}\|_\infty \\ \text{s.t.} \quad & u_i \geq 0, \quad \forall i. \end{aligned} \tag{16}$$

Next we show we can get rid of the nonnegative constraints and derive the optimal solution to (16) by the optimal solution to the unconstrained version:

$$\min_{\mathbf{u}} \quad \frac{1}{2}\|\mathbf{u} - \mathbf{b}\|_2^2 + \frac{\alpha}{L}\|\mathbf{u}\|_\infty. \tag{17}$$

First, we need the following lemma.

Lemma 1. *Let $\mathbf{u}^*$ be the optimal solution to (17). For any element u_i^* of $\mathbf{u}^*$, we have*

$$\begin{aligned} b_i = 0 &\Rightarrow u_i^* = 0, \\ b_i > 0 &\Rightarrow u_i^* \geq 0, \\ b_i < 0 &\Rightarrow u_i^* \leq 0. \end{aligned}$$

The proof can be found in the appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2015.2448542. Lemma 1 indicates that (1) if all the elements of $\mathbf{b}$ are nonnegative, we can remove the nonnegative constraints in (16) safely; (2) we can analyze the optimal solution to (16) by dealing with positive and negative elements in $\mathbf{b}$ separately. The following proposition gives the optimal solution to (16).

Proposition 1. *Let $\tilde{\mathbf{b}}$ be a vector consisting of the positive elements in $\mathbf{b}$ and $\tilde{\mathbf{u}}$ represent a variable vector consisting of the elements in $\mathbf{u}$ at the same indices. Let $\tilde{\mathbf{u}}^*$ be the optimal solution to $\min_{\tilde{\mathbf{u}}}\frac{1}{2}\|\tilde{\mathbf{u}} - \tilde{\mathbf{b}}\|_2^2 + \frac{\alpha}{L}\|\tilde{\mathbf{u}}\|_\infty$. The optimal solution $\mathbf{u}^*$ to (16) can be obtained as follows: for each index i, if $b_i \leq 0$, then $u_i^* = 0$; otherwise, u_i^* is obtained as the corresponding element of $\tilde{\mathbf{u}}^*$.*

The proof is in the appendix, available in the online supplemental material. By Proposition 1, we just need to extract positive elements from $\mathbf{b}$ to form $\tilde{\mathbf{b}}$ and address the unconstrained problem:

$$\min_{\tilde{\mathbf{u}}} \quad \frac{1}{2}\|\tilde{\mathbf{u}} - \tilde{\mathbf{b}}\|_2^2 + \frac{\alpha}{L}\|\tilde{\mathbf{u}}\|_\infty. \tag{18}$$

Although (18) can be addressed using properties of the subgradient set of $\|\tilde{\mathbf{u}}\|_\infty$, a more efficient solution to its dual form is well established. Define $\mathbf{a}$ as the dual variable. We have

$$\begin{aligned} &\min_{\tilde{\mathbf{u}}}\left[\frac{1}{2}\|\tilde{\mathbf{u}} - \tilde{\mathbf{b}}\|_2^2 + \frac{\alpha}{L}\|\tilde{\mathbf{u}}\|_\infty\right] \\ &= \min_{\tilde{\mathbf{u}}}\max_{\mathbf{a}}\left[\mathbf{a}^T(\tilde{\mathbf{b}} - \tilde{\mathbf{u}}) - \frac{1}{2}\|\mathbf{a}\|_2^2 + \frac{\alpha}{L}\|\tilde{\mathbf{u}}\|_\infty\right] \\ &= \max_{\mathbf{a}}\min_{\tilde{\mathbf{u}}}\left[-\mathbf{a}^T\tilde{\mathbf{u}} + \frac{\alpha}{L}\|\tilde{\mathbf{u}}\|_\infty + \mathbf{a}^T\tilde{\mathbf{b}} - \frac{1}{2}\|\mathbf{a}\|_2^2\right] \\ &= \max_{\mathbf{a}}\left[\mathbf{a}^T\tilde{\mathbf{b}} - \frac{1}{2}\|\mathbf{a}\|_2^2 \quad s.t. \|\mathbf{a}\|_1 \leq \frac{\alpha}{L}\right], \end{aligned}$$

which is equivalent to the following problem

$$\min_{\mathbf{a}}\frac{1}{2}\|\mathbf{a} - \tilde{\mathbf{b}}\|_2^2 \quad s.t. \|\mathbf{a}\|_1 \leq \frac{\alpha}{L}. \tag{19}$$

Moreover, $\mathbf{a}$ satisfies the relation $\mathbf{a} = \tilde{\mathbf{b}} - \tilde{\mathbf{u}}$. Thus, by solving (19) we can readily obtain a solution for (18). (19) is the "Euclidean projection onto the L_1-ball" problem and an efficient solution has been proposed by Duchi et al. [34]. Here we only describe the results but skip the analysis. In our case the optimal vector $\mathbf{a}^*$ is computed as

$$a_i^* = \max(0, \tilde{b}_i - \theta), \tag{20}$$

where θ is a nonnegative scalar. Let $\tilde{b}_{(i)}$ be the ith largest element in $\tilde{\mathbf{b}}$. θ is computed as

$$\theta = \frac{1}{\rho}\left(\sum_{i=1}^{\rho} \tilde{b}_{(i)} - \frac{\alpha}{L}\right), \tag{21}$$

where ρ is the maximum index satisfying the condition

$$\sum_{i=1}^{\rho}(\tilde{b}_{(i)} - \tilde{b}_{(\rho)}) < \frac{\alpha}{L}. \tag{22}$$

In order to handle the smallest element in $\tilde{\mathbf{b}}$, an extra 0 is added to $\tilde{\mathbf{b}}$, i.e., $\tilde{b}_{(M^{+}+1)} = 0$ where M^{+} is the length of $\tilde{\mathbf{b}}$. If we test up to the 0 element to find $\sum_{i=1}^{M^{+}+1}(\tilde{b}_{(i)} - 0) = \sum_{i=1}^{M^{+}} \tilde{b}_i < \frac{\alpha}{L}$, the optimal choice for θ is 0. In this case, $\tilde{\mathbf{b}}$ is in the L_1-ball and $\mathbf{a}^{*} = \tilde{\mathbf{b}}$. This corresponds to $\tilde{\mathbf{u}}^{*} = \mathbf{0}$, i.e., the corresponding basis vector is zeroed-out.

The key of the optimization of (19) is calculating θ. A straightforward way is to sort $\tilde{\mathbf{b}}$ and process the elements sequentially. In this work, we employ the algorithm introduced by Duchi et al. [34] to calculate θ which is linear in M^{+}. The algorithm is presented in the appendix, available in the online supplemental material.

4.2 Optimizing V

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

$$\begin{aligned} \min_{\mathbf{V}} \ \psi(\mathbf{V}) := \Bigg(&\frac{1}{2}\sum_{v=1}^{H}\|\mathbf{X}^{(v)} - \mathbf{U}^{(v)}\mathbf{V}\|_F^2 + \gamma\|\mathbf{V}\|_{1,1} \\ &+ \frac{\beta}{2}\{tr[\mathbf{V}^l\mathbf{L}^a(\mathbf{V}^l)^T] - tr[\mathbf{V}^l\mathbf{L}^p(\mathbf{V}^l)^T]\}\Bigg) \\ \text{s.t.} \quad & 1 \geq V_{kj} \geq 0, \ \ \forall j,k. \end{aligned} \tag{23}$$

This is a bounded nonnegative quadratic programming problem for $\mathbf{V}$. Sha et al. [35] proposed a general multiplicative optimization scheme for this kind of problems. We follow their idea and develop a multiplicative update algorithm for optimizing $\mathbf{V}$. The minor difference is that they dealt with vector variables while in our case the variable is a matrix.

First, 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 $\psi(\mathbf{V})$:

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

For compactness and clarity, 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. (23) can be transformed into

$$\begin{aligned} \min_{\mathbf{V}} \quad & \frac{1}{2}tr[(\mathbf{V}^l)^T\mathbf{P}\mathbf{V}^l] - tr[(\mathbf{V}^l)^T\mathbf{Q}^l] + \gamma\|\mathbf{V}^l\|_{1,1} \\ & + \frac{\beta}{2}\{tr[\mathbf{V}^l\mathbf{L}^a(\mathbf{V}^l)^T] - tr[\mathbf{V}^l\mathbf{L}^p(\mathbf{V}^l)^T]\} \\ & + \frac{1}{2}tr[(\mathbf{V}^u)^T\mathbf{P}\mathbf{V}^u] - tr[(\mathbf{V}^u)^T\mathbf{Q}^u] + \gamma\|\mathbf{V}^u\|_{1,1} \\ \text{s.t.} \quad & 1 \geq V_{kj} \geq 0, \ \ \forall j,k. \end{aligned} \tag{24}$$

Therefore, we can update $\mathbf{V}^l$ and $\mathbf{V}^u$ separately. To derive the update rules, similar auxiliary functions as those in [35] can be designed. We only show the results here and defer the detailed derivations to the appendix, available in the online supplemental material:

$$V_{kj}^l \leftarrow \min\left\{1, \frac{-B_{kj} + \sqrt{B_{kj}^2 + 4A_{kj}C_{kj}}}{2A_{kj}}V_{kj}^l\right\}, \tag{25}$$

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

where A_{kj}, B_{kj} and C_{kj} are

$$A_{kj} = (\mathbf{P}\mathbf{v}_j^l)_k + \beta((\mathbf{D}^a + \mathbf{W}^p)\bar{\mathbf{v}}_k^l)_j, \tag{27}$$

$$B_{kj} = \gamma - Q_{kj}^l, \tag{28}$$

$$C_{kj} = \beta((\mathbf{D}^p + \mathbf{W}^a)\bar{\mathbf{v}}_k^l)_j. \tag{29}$$

Here $\mathbf{v}_j^l$ and $\bar{\mathbf{v}}_k^l$ denote the jth column vector and the kth row vector of $\mathbf{V}^l$, respectively.

4.3 Computational Complexity Analysis

The major space cost of MCL is due to the matrices $\{\mathbf{U}^{(v)}\}_{v=1}^{H}$, $\mathbf{V}$, $\mathbf{W}^a$ and $\mathbf{W}^p$, which is $O(K(\sum_{v=1}^{H} M_v + N) + 2(N^l)^2)$. The time complexity consists of two parts, corresponding to the subproblems for $\{\mathbf{U}^{(v)}\}_{v=1}^{H}$ and $\mathbf{V}$ respectively. For optimizing each $\mathbf{U}^{(v)}$, we need to run Algorithm 2. The core step is the optimization of (14), which requires solving (16) for each column of $\mathbf{U}^{(v)}$. The cost of solving (16) for a column of $\mathbf{U}^{(v)}$ is $O(M_v)$, so the total cost of solving (14) is $O(M_vK)$. In the outer loop of Algorithm 2 we also need to compute $\nabla f(\mathbf{U})$ ($O(M_vK^2)$ if we pre-compute $\mathbf{V}\mathbf{V}^T$ and $\mathbf{X}^{(v)}\mathbf{V}^T$). Assume we run T iterations of the outer loop of Algorithm 2, and recall that the total number of iterations of the inner loop is upper bounded by $2(T+1) + \log_2\frac{L_f}{L_0}$. The major cost for optimizing $\mathbf{U}^{(v)}$ is $O(M_vK(2(T+1) + log_2\frac{L_f}{L_0}) + TM_vK^2)$. Regarding $\mathbf{V}$, we update $\mathbf{V}^l$ and $\mathbf{V}^u$ iteratively. Let N^l and N^u be the number of labeled and unlabeled items, respectively. In each iteration, we need to compute three matrices for $\mathbf{V}^l$ (Eqs. (27)-(29)): $\mathbf{A}$ ($O(N^lK^2 + (N^l)^2K)$), $\mathbf{B}$ ($O(N^lK)$) and $\mathbf{C}$ ($O((N^l)^2K)$). For $\mathbf{V}^u$, the cost for computing $\mathbf{P}\mathbf{V}^u$ is $O(N^uK^2)$. Combining these pieces together and incorporating the costs of Eqs. (25) and (26), the major cost for optimizing $\mathbf{V}$ is

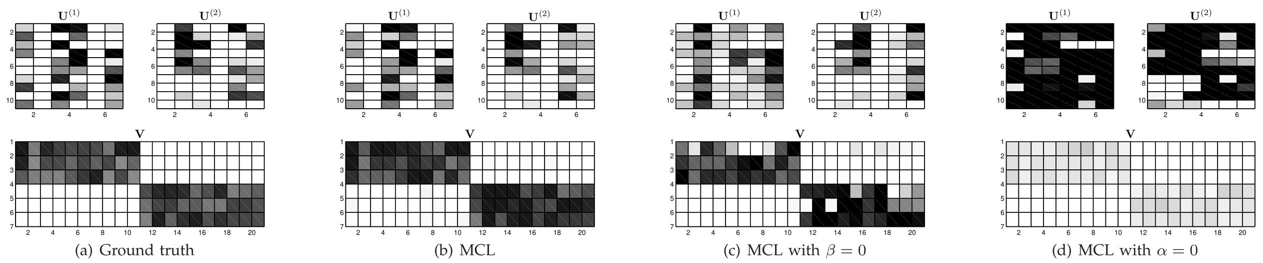


Fig. 2. Latent spaces recovered on a toy example. (a) The true underlying basis matrices and consensus encoding matrix generated randomly as described in the text. (b) The latent space recovered by MCL. (c) The latent space recovered by MCL with $\beta = 0$, i.e., removing influence of the graph embedding terms. (d) The latent space recovered by MCL with $\alpha = 0$, i.e., removing influence of the $L_{1,\infty}$ penalty terms on $\mathbf{U}$'s. Fully white columns correspond to zero-valued vectors. For $\mathbf{V}$, we let 1 correspond to fully black. For $\mathbf{U}$'s, 1.5 corresponds to fully black and all values above 1.5 are drawn as fully black.

$O(T'(NK + NK^2 + (N^l)^2 K))$, where T' is the number of iterations. We can see that the time cost of MCL is linear in N and M_v, if other variables are fixed. Although $(N^l)^2$ is a quadratic term, it only depends on the number of labeled items. In practice, labeled items are far less than unlabeled items. Considering variables other than N and M_v are typically much smaller, MCL is efficient.

In practice, we do not have to optimize the subproblems for $\{\mathbf{U}^{(v)}\}_{v=1}^{H}$ and $\mathbf{V}$ until convergence in each iteration of Algorithm 1. Instead, we optimize each subproblem for a few iterations, which also prevents the optimization from being too greedy. In experiments, we set $T = T' = 10$.

5 Experiments

In this section, we show the empirical results of our approach on learning latent representations from multi-view inputs. We first applied MCL on a synthetic toy problem to investigate the efficacy of the proposed optimization method. Then three real-world datasets were employed for quantitatively testing MCL's performance on item classification and clustering.

5.1 A Toy Example

The major properties of MCL include: (1) The semantic relationships between items can be captured by the learned latent space through incorporating label information by a graph embedding framework. (2) Each dimension of the learned latent space has the flexibility of being associated with an arbitrary subset of views, which is encouraged by the $L_{1,\infty}$ terms. We constructed a toy factorization problem to investigate whether these properties can be recovered by the proposed optimization method. The toy dataset consisted of two views of 20 data items generated from two categories, c_1 and c_2. The first 10 items belonged to c_1 and the remaining ones belonged to c_2. The underlying latent space had six dimensions, with three for each category. Among the three dimensions for a category, two dimensions were private to view 1 and view 2 respectively, and the third one was a shared dimension. The dimensionality of both views was 10. First, we randomly generated 10×6 basis matrices for the two views, with elements produced by a Gamma distribution $\mathrm{Gamma}(1, 0.9)$. We also randomly retained 30-40 percent elements of each basis vector to be zero to simulate the notion of "parts". Second, the consensus encoding matrix was generated, where each item was a weighted combination of the basis vectors for the corresponding category and the weights were randomly distributed in $(0.4, 1)$. Finally, we obtained the data matrices via multiplying the generated basis matrices by the consensus encoding matrix and adding Gaussian noise with standard deviation 0.05. The generated basis matrices and consensus encoding matrix are shown in Fig. 2a, where the first three columns of $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are for c_1 and the remaining ones are for c_2.

To run MCL, we also need to provide partial label information for the dataset. We treated five items from each category as labeled items. Fig. 2b shows the latent space recovered by MCL. We can see that MCL correctly factorized the data matrices, capturing both the semantic structure in the encoding matrix and the shared-private structure in the basis matrices. The recovered results were very similar to the ground truth. This indicates the two important properties discussed above can be recovered by our optimization method. We further tested the necessity of the regularization terms. First, we set $\beta = 0$, which corresponded to removing the graph embedding terms. The results are shown in Fig. 2c. We found in this setting the semantic structure of $\mathbf{V}$ was not well recovered, e.g. some items of c_2 had weak connections with c_1's latent features. The recovered $\mathbf{V}$ exhibited a larger within-class variance and a lower average between-class distance. As a side effect, $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ were not well recovered either. Second, we removed the influence of the $L_{1,\infty}$ terms by setting $\alpha = 0$[1]. The learned latent space is shown in Fig. 2d. Without the $L_{1,\infty}$ constraints, the optimization algorithm tended to learn dense basis matrices with a lot of large values (some above 10). Consequently, the weights in $\mathbf{V}$ became much lower, although the semantic structure was still preserved.

To summarize, we demonstrated via a toy example that (1) the optimization method worked as expected; (2) both the graph embedding regularization and the structured sparseness regularization were important for recovering the conceptual structures in the underlying latent space.

5.2 Experiments on Real-World Datasets

5.2.1 Datasets

We use three real-world datasets to evaluate MCL:

Reuters. This dataset was constructed from the Reuters Multilingual collection [36] which contains totally 111,740 news documents written in five different languages.

1. In this case we also set $\gamma = 0$ to eliminate the scaling issue for non-negative sparse coding models.

TABLE 1
Statistics of the Datasets

Dataset	Size	# of categories	Dimensionality of views
Reuters	1,800	6	21,531/15,506/11,547
MM2.0	5,000	25	64/144/75/128
ImgNet	10,000	50	64/1000/512

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. For each category, we randomly sampled 300 documents, resulting in a dataset with 1,800 documents in total.

MM2.0. The second dataset came from Microsoft Research Asia Internet Multimedia Dataset 2.0 (MSRA-MM 2.0) [37]. MSRA-MM 2.0 consists of about 1 million images which were respective search results for 1,165 popular query concepts in Microsoft Live Search. Each concept has approximately 500-1,000 images. For each image, its relevance to the corresponding concept was manually labeled with three levels: very relevant, relevant and irrelevant. Seven low level features were extracted for each image. To form the experimental dataset, we randomly selected 25 query concepts from the *Animal*, *Object* and *Scene* branches as categories, and then selected 200 images for each concept. Specifically, if the concept had more than 200 "very relevant" images, we simply randomly sampled 200 images from them. Otherwise, all "very relevant" images were taken and, if less than 200, we further randomly sampled images from "relevant" images. "Irrelevant" images were discarded. We took four features in MSRA-MM 2.0 as four views: 64D HSV color histogram, 144D color correlogram, 75D edge distribution histogram and 128D wavelet texture. *ImgNet.* The third dataset was obtained from ImageNet [38], a real-world image database containing roughly 15 million images organized according to the WordNet hierarchy. Currently, over 20 thousand noun synsets in WordNet are indexed and each synset has over 500 images on average. We randomly chose 50 leaf synsets in the hierarchy as categories and randomly sampled 200 images from each selected synset. Three views of this dataset were 64D HSV histogram, 1,000D bag of SIFT [39] visual words, and 512D GIST descriptors [40].

The statistics of these datasets are summarized in Table 1.

5.2.2 Evaluation Methodology

In order to show the effectiveness of MCL, we compared it with the following baseline methods:

- NMF on best view (*NMF-b*): This baseline simply applies NMF [21] on each view and reports the best performance.
- Feature concatenation (*ConcatNMF*): This method concatenates feature vectors of different views to form a united representation and then applies NMF.
- Multi-view NMF (*MultiNMF*): This is the unsupervised Multi-view NMF algorithm proposed in [17].
- Semi-supervised Unified Latent Factor method (*SULF*): SULF [3] is a multi-view nonnegative factorization method which models partial label information as a factorization constraint on $\mathbf{V}^l$.
- Graph regularized NMF (*GNMF*): This variant of NMF was originally proposed as a manifold regularized version of NMF [41]. 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. (6) to make it a semi-supervised method on multi-view data.

Note that the first three are unsupervised methods while the last two are semi-supervised methods.

The above six methods were evaluated by text/image classification and clustering. We adopted an evaluation scheme similar to 5×2 cross-validation [42], [43]. For each dataset, we generated five random train-test splits. In a split, we randomly took 50 percent items from each category as labeled training data and put the remaining items into the test set. Two-fold cross-validation was performed for each split. Thus, there were totally 10 test cases. For each test case, we run each method three times and computed the averaged performance. The overall averaged performance and standard deviation were reported. In case the method has parameters, we tuned the parameters on a separate random split. The dimensionalities of the latent space were empirically set to 50, 100 and 150, for Reuters, MM2.0 and ImgNet respectively.

We used the learned representations of different methods for classification and clustering. For classification, the training items were fed to a kNN classifier ($k = 9$) and the *Accuracy* of the classifier on the test items was calculated. For clustering, k-means was employed as the clustering method. Since the semi-supervised methods made use of the label information of training data, we only performed clustering on test items for fairness. To measure clustering performance, two metrics, *Accuracy* and *Normalized Mutual Information* (NMI) were used. Accuracy is defined as

$$\text{Accuracy} = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n}, \tag{30}$$

where n is the total number of items for clustering, r_i and s_i are cluster labels of item i in clustering results and in ground truth, respectively, $\delta(x, y)$ equals 1 if $x = y$ and equals 0 otherwise, and $map(r_i)$ is the permutation mapping function which maps r_i to the equivalent cluster label in ground truth. This mapping can be obtained by the Kuhn-Munkres algorithm [44]. Given two sets of item clusters C and $C^\dagger$, NMI is defined as

$$\text{NMI}(C, C^\dagger) = \frac{MI(C, C^\dagger)}{\max(H(C), H(C^\dagger))}, \tag{31}$$

where $H(C)$ denotes the entropy of cluster set C. $MI(C, C^\dagger)$ is the mutual information between C and $C^\dagger$:

$$MI(C, C^\dagger) = \sum_{c_i \in C, c_j^\dagger \in C^\dagger} p(c_i, c_j^\dagger) \log_2 \frac{p(c_i, c_j^\dagger)}{p(c_i) p(c_j^\dagger)}. \tag{32}$$

$p(c_i)$ is the probability that a randomly selected item from all testing items belongs to cluster c_i, and $p(c_i, c_j^\dagger)$ is the joint

TABLE 2
Classification Performance of Different Factorization Methods on the Reuters Dataset (Accuracy $\pm$ std dev,%)

Labeled Percentage	NMF-b	ConcatNMF	MultiNMF	SULF	GNMF	MCL
10	61.90 ± 3.91	62.81 ± 3.77	63.51 ± 3.41	64.58 ± 2.54	**68.13 ± 1.91**	67.97 ± 1.79
20	65.50 ± 2.44	65.97 ± 2.40	67.33 ± 2.14	68.35 ± 1.87	69.53 ± 1.79	**71.78 ± 1.99**
30	67.15 ± 2.18	68.38 ± 2.18	69.58 ± 1.63	69.88 ± 1.71	70.80 ± 1.22	**73.47 ± 1.61**
40	68.87 ± 1.76	69.33 ± 1.94	70.38 ± 1.72	70.71 ± 1.15	72.04 ± 1.74	**74.43 ± 1.42**
50	69.58 ± 1.71	70.30 ± 2.03	71.25 ± 1.43	72.02 ± 1.42	73.06 ± 1.45	**76.02 ± 1.10**

TABLE 3
Classification Performance of Different Factorization Methods on the MM2.0 Dataset (Accuracy $\pm$ std dev, %)

Labeled Percentage	NMF-b	ConcatNMF	MultiNMF	SULF	GNMF	MCL
10	23.10 ± 1.23	27.38 ± 0.95	26.25 ± 1.21	27.21 ± 1.19	27.91 ± 1.18	**30.26 ± 1.14**
20	25.24 ± 1.04	31.10 ± 1.10	30.20 ± 1.32	30.53 ± 1.23	31.48 ± 1.23	**34.05 ± 1.08**
30	26.98 ± 0.86	32.22 ± 0.92	31.74 ± 0.87	32.80 ± 0.91	33.82 ± 0.85	**36.33 ± 0.92**
40	28.13 ± 1.08	34.55 ± 0.87	33.66 ± 0.85	34.73 ± 0.82	35.40 ± 1.07	**37.38 ± 0.81**
50	28.69 ± 0.79	35.42 ± 1.14	34.73 ± 0.70	36.13 ± 0.58	36.71 ± 0.70	**38.48 ± 0.75**

TABLE 4
Classification Performance of Different Factorization Methods on the ImgNet Dataset (Accuracy $\pm$ std dev,%)

Labeled Percentage	NMF-b	ConcatNMF	MultiNMF	SULF	GNMF	MCL
10	12.90 ± 1.23	17.15 ± 1.03	16.37 ± 0.91	19.95 ± 1.12	**21.89 ± 0.99**	21.38 ± 1.06
20	14.54 ± 1.13	20.03 ± 0.91	20.29 ± 0.66	22.55 ± 0.85	24.41 ± 0.85	**25.77 ± 0.89**
30	15.93 ± 0.85	22.07 ± 0.84	22.31 ± 0.50	23.79 ± 0.62	25.82 ± 1.14	**28.31 ± 0.90**
40	17.21 ± 0.80	23.28 ± 0.71	23.79 ± 0.56	24.37 ± 0.77	26.59 ± 0.83	**30.10 ± 0.74**
50	18.08 ± 0.87	24.32 ± 0.73	24.59 ± 0.49	25.29 ± 0.68	27.38 ± 0.84	**31.09 ± 0.67**

probability that a randomly selected item is in c_i and $c_j^\dagger$ simultaneously. If C and $C^\dagger$ are identical, $\mathrm{NMI}(C, C^\dagger) = 1$. $\mathrm{NMI}(C, C^\dagger) = 0$ when the two cluster sets are completely independent. So the range of NMI is $[0, 1]$. In our case, C represents obtained clusters and $C^\dagger$ is the ground truth.

5.2.3 *Performance Comparison*

Tables 2, 3 and 4 show the classification performance results on Reuters, MM2.0 and ImgNet, respectively. We varied the percentage of training items from 10 to 50 percent. Observations are as follows. First, methods that made use of multiple features (i.e., views) of items outperformed NMF-b, which only exploited 1 view. This is in accord with the results of previous multi-view learning work. Second, semi-supervised methods (the last three columns) tended to outperform unsupervised methods, which indicated that exploiting label information could lead to latent spaces with better discriminative structures. Third, MCL and GNMF often showed superior performance over SULF. As aforementioned in Section 2.2, SULF models label information as a factorization constraint on $\mathbf{V}^l$, i.e., reconstructing the label indicator vector of item i by multiplying its encoding $\mathbf{v}_i^l$ by a weight matrix. Although identical encoding vectors lead to identical label indicator vectors, it is also possible to reconstruct the label indicator vector by combining columns of the weight matrix in different ways. Therefore, this can be viewed as imposing indirect affinity constraints on encodings of within-class items. On the contrary, the graph embedding terms in MCL and GNMF impose direct affinity constraints on item encodings and therefore could lead to clearer conceptual structures in the learned latent spaces. Finally, MCL outperformed the baseline methods under almost all cases. We performed F-test for 5×2 cross-validation [45] with significance level 0.05. The results indicated that MCL was significantly superior over all baselines except for the 10 percent cases of Reuters and ImgNet. MCL not only exploits label information via a graph embedding framework, but also allows flexible latent factor sharing among different views by encouraging each $\mathbf{U}^{(v)}$ to be sparse in columns. These properties could help learn a more meaningful conceptual latent space. In Section 5.2.4, we will present parameter study and demonstrate that the semi-supervised terms and the sparseness terms indeed contribute to the performance of MCL.

The clustering results are shown in Figs. 3, 4 and 5, for Reuters, MM2.0 and ImgNet respectively. The observations were very similar to those for classification. According to

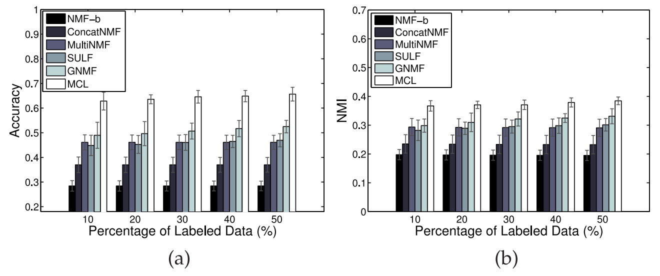


Fig. 3. Clustering performance of different methods on Reuters. Error bars represent standard deviations.

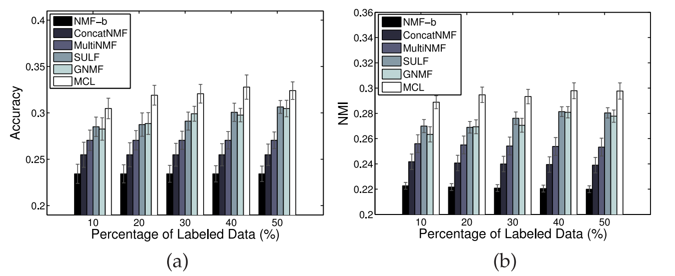


Fig. 4. Clustering performance of different methods on MM2.0. Error bars represent standard deviations.

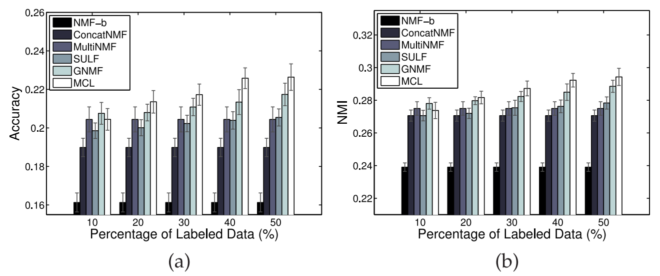


Fig. 5. Clustering performance of different methods on ImgNet. Error bars represent standard deviations.

F-test with significance level 0.05, we found MCL significantly outperformed the baseline methods under almost all cases (except the 10 percent case of Accuracy and 10 and 20 percent cases of NMI on ImgNet, in Fig. 5).

5.2.4 *Parameter Study*

MCL has three parameters, α, β and γ. β measures the importance of the semi-supervised part of MCL (i.e., the graph embedding regularization terms), while α and γ control the degree of sparsity of the basis matrices and the consensus encoding matrix respectively. We investigated their influence on MCL's performance by varying one parameter at a time while fixing the other two. For each specific setting, we run MCL 20 times and the average performance was recorded.

The results are shown in Figs. 6 and 7 for Reuters and MM2.0 respectively (results for ImgNet were similar with those for MM2.0). We found the general behavior of the three 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 space. Specific observations are as follows. First, the model's performance was not very sensitive to the value of α ($L_{1,\infty}$ is intrinsically a summation of only K terms). MCL achieved its best performance when α was in $[150, 250]$ and $[50, 200]$, for Reuters and MM2.0 respectively. Second, β's impact in Reuters appeared to be smaller than its impact in image datasets. This could be because that in Reuters α dominated the performance boost (e.g. the clustering accuracy increased by about 15 percent when varying α, as shown in Fig. 6a). Finally, γ seemed to contribute little to the performance of MCL in MM2.0. As shown by Fig. 7c, the clustering performance degenerated when increasing γ from 0, although the classification accuracy increased a bit at the beginning. Based on these observations, we set $\alpha = 150$, $\beta = 0.02$ and $\gamma = 0.005$ for other experiments.

5.2.5 *Convergence Analysis*

The optimization method for MCL solves the subproblems for $\{\mathbf{U}^{(v)}\}_{v=1}^{H}$ and $\mathbf{V}$ iteratively to find a local minimum of (10). Here we analyze its empirical convergence properties.

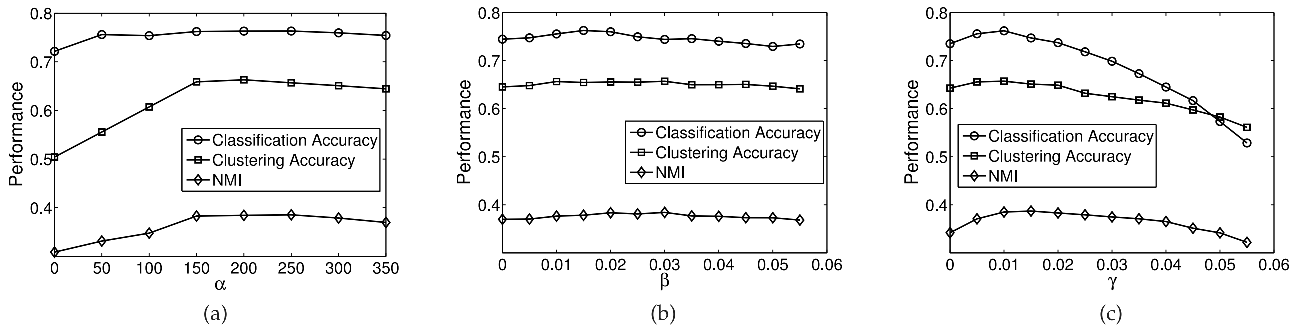


Fig. 6. Influence of different parameter settings on the performance of MCL in the Reuters dataset: (a) varying α while setting $\beta = 0.02$ and $\gamma = 0.005$, (b) varying β while setting $\alpha = 150$ and $\gamma = 0.005$, and (c) varying γ while setting $\alpha = 150$ and $\beta = 0.02$.

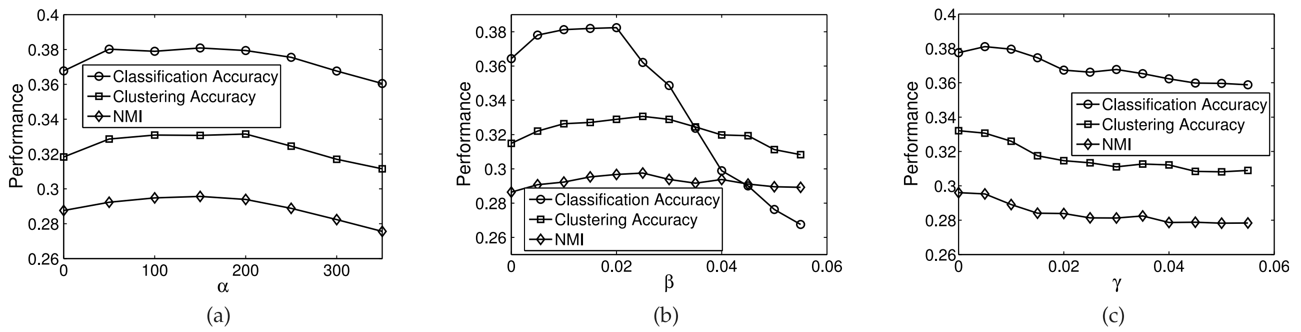


Fig. 7. Influence of different parameter settings on the performance of MCL in the MM2.0 dataset: (a) varying α while setting $\beta = 0.02$ and $\gamma = 0.005$, (b) varying β while setting $\alpha = 150$ and $\gamma = 0.005$, and (c) varying γ while setting $\alpha = 150$ and $\beta = 0.02$.

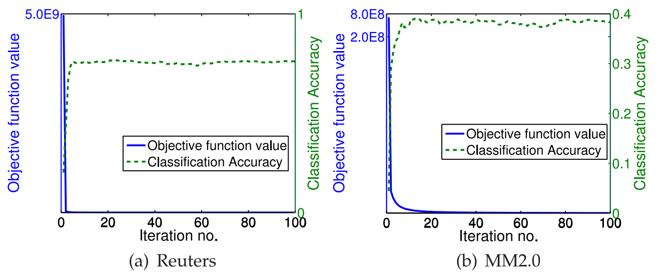


Fig. 8. Convergence analysis of MCL on (a) Reuters and (b) MM2.0. The *y*-axes for objective function values are in log scale.

Figs. 8a and 8b plot both the objective function value and the classification accuracy against the number of iterations performed, for Reuters and MM2.0 respectively. Its behavior on ImgNet was similar. We found that at the beginning the objective function value dropped drastically and the performance increased rapidly. The converging speed was faster on text data than on image data. The reason would be that text data has clearer conceptual structures than image data. The optimization procedure typically converged around 40 iterations and 100 iterations, for Reuters and MM2.0 respectively. Although the converging speed was not very fast, the performance achieved the best very fast. We can see from Fig. 8 that the performance becomes stable in about 10 iterations for Reuters and 20 for MM2.0.

6 Conclusions

In this work we proposed Multi-view Concept Learning, a novel nonnegative latent representation learning algorithm for representation learning from multi-view data. MCL tried to learn a conceptual latent space 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 one another and kept items belonging to different categories as distant as possible, in the learned latent space. Another novel property of MCL was that it allowed each latent dimension to be associated with a subset of views by imposing $L_{1,\infty}$ regularization on each basis matrix $\mathbf{U}^{(v)}$. Therefore, MCL is able to learn flexible latent factor sharing structures which could lead to more meaningful conceptual latent spaces. We proposed an efficient optimization method for MCL and demonstrated its efficacy by a toy factorization problem. We used two real-world datasets to evaluate the empirical performance of MCL. Experimental results indicated that MCL was effective and outperformed baseline methods.

The proposed graph embedding framework is a general framework in that different definitions of the within-class affinity graph G^a and the between-class penalty graph G^p can be employed. For example, we could (1) customize the weights in $\mathbf{W}^a$ and $\mathbf{W}^p$ according to the similarities between items; (2) employ maximum margin style definitions [22]; (3) incorporate data manifolds [41], [46], [47] to better connect labeled items and unlabeled items; etc. However, the focus of this work was to propose a conceptual latent space learning algorithm based on multi-view data and demonstrate its effectiveness. We leave these possible improvements to future work.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grant No. 61373118, the National Basic Research Program of China (973 Program) under Grant 2013CB336500, and the Program for Changjiang Scholars and Innovative Research Team in University (No. IRT1309). The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

[1] I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel, “Social media recommendation based on people and tags,” in *Proc. 33rd Int. SIGIR Conf. Res. Develop. Inf. Retrieval*, 2010, pp. 194–201.
[2] Y. Han, F. Wu, D. Tao, J. Shao, Y. Zhuang, and J. Jiang, “Sparse unsupervised dimensionality reduction for multiple view data,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 10, pp. 1485–1496, Oct. 2012.
[3] Y. Jiang, J. Liu, Z. Li, and H. Lu, “Semi-supervised unified latent factor learning with Multi-view data,” *Mach. Vis. Appl.*, vol. 25, pp. 1635–1645, 2013.
[4] S. Romberg, R. Lienhart, and E. Hörster, “Multimodal image retrieval,” *Int. J. Multimedia Inf. Retrieval*, vol. 1, no. 1, pp. 31–44, 2012.
[5] F. Korn, B.-U. Pagel, and C. Faloutsos, “On the dimensionality curse and the self-similarity blessing,” *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 1, pp. 96–111, Jan./Feb. 2001.
[6] W. Wang and Z.-H. Zhou, “A new analysis of Co-training,” in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 1135–1142.
[7] S. Bickel and T. Scheffer, “Multi-view clustering,” in *Proc. 4th IEEE Int. Conf Data Mining*, 2004, pp. 19–26.
[8] D. Zhou and C. J. Burges, “Spectral clustering and transductive learning with multiple views,” in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 1159–1166.
[9] M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy, “Composite kernel learning,” *Mach. Learn.*, vol. 79, no. 1–2, pp. 73–103, 2010.
[10] Z. Xu, R. Jin, H. Yang, I. King, and M. R. Lyu, “Simple and efficient multiple kernel learning by group lasso,” in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 1175–1182.
[11] M. Chen, K. Q. Weinberger, and J. Blitzer, “Co-training for domain adaptation,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2456–2464.
[12] Z. Xu and S. Sun, “Multi-view transfer learning with adaboost,” in *Proc. 23rd IEEE Int. Conf. Tools Artif. Intell.*, 2011, pp. 399–402.
[13] H. Hotelling, “Relations between two sets of variates,” *Biometrika*, vol. 28, pp. 321–377, 1936.
[14] Y. Jia, M. Salzmann, and T. Darrell, “Factorized latent spaces with structured sparsity,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 982–990.
[15] B. Long, S. Y. Philip, and Z. M. Zhang, “A general model for multiple view unsupervised learning,” in *Proc Int. Conf. Data Mining*, 2008, pp. 822–833.
[16] M. Kalayeh, H. Idrees, and M. Shah, “Nmf-knn: Image annotation using weighted multi-view non-negative matrix factorization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 184–191.
[17] J. Liu, C. Wang, J. Gao, and J. Han, “Multi-view clustering via joint nonnegative matrix factorization,” in *Proc. SIAM Int. Conf. Data Mining*, 2013, vol. 13, pp. 252–260.
[18] T. Xia, D. Tao, T. Mei, and Y. Zhang, “Multiview spectral embedding,” *IEEE Trans. Syst., Man Cybern., Part B: Cybern.*, vol. 40, no. 6, pp. 1438–1446, Dec. 2010.
[19] N. Chen, J. Zhu, and E. P. Xing, “Predictive subspace learning for multi-view data: A large margin approach,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 361–369.
[20] A. Shon, K. Grochow, A. Hertzmann, and R. P. Rao, “Learning shared latent structure for image synthesis and robotic imitation,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1233–1240.
[21] D. D. Lee and H. S. Seung, “Learning the parts of objects by Non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[22] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[23] Y. Nesterov, "Gradient methods for minimizing composite functions," *Math. Programm.*, vol. 140, no. 1, pp. 125–161, 2013.

[24] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1336–1353, Jun. 2013.

[25] Y. Wang, Y. Jia, C. Hu, and M. Turk, "Fisher Non-negative matrix factorization for learning local features," in *Proc. Asian Conf. Vis.*, 2004. pp. 27–30.

[26] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas, "Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 683–695, May 2006.

[27] P. O. Hoyer, "Non-negative sparse coding," in *Proc. 12th IEEE Workshop Neural Netw. Signal Process.*, 2002, pp. 557–565.

[28] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating Non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.

[29] N. Mohammadiha and A. Leijon, "Nonnegative matrix factorization using projected gradient algorithms with sparseness constraints," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol.*, 2009, pp. 418–423.

[30] C. Xu, D. Tao, and C. Xu, "A survey on Multi-view learning," *arXiv preprint arXiv:1304.5634*, 2013.

[31] A. Quattoni, X. Carreras, M. Collins, and T. Darrell, "An efficient projection for l 1, infinity regularization," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 857–864.

[32] I. Kotsia, S. Zafeiriou, and I. Pitas, "A novel discriminant Non-negative matrix factorization algorithm with applications to facial image characterization problems," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3–2, pp. 588–595, Sep. 2007.

[33] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, 2007.

[34] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l 1-ball for learning in high dimensions," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 272–279.

[35] F. Sha, Y. Lin, L. K. Saul, and D. D. Lee, "Multiplicative updates for nonnegative quadratic programming," *Neural Comput.*, vol. 19, no. 8, pp. 2004–2031, 2007.

[36] M. Amini, N. Usunier, and C. Goutte, "Learning from multiple partially observed Views-an application to multilingual text categorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 28–36.

[37] H. Li, M. Wang, and X.-S. Hua, "Msra-mm 2.0: A Large-scale web multimedia dataset," in *Proc. IEEE Int. Conf. Data Mining Workshops*, 2009, pp. 164–169.

[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A Large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.

[39] D. G. Lowe, "Distinctive image features from Scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[40] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[41] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[42] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, 1998.

[43] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

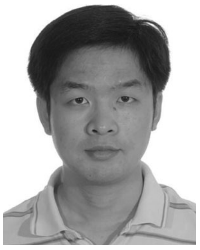
[44] L. Lovasz and M. D. Plummer, *Matching Theory*. Amsterdam, The Netherlands: North Holland, 1986.

[45] E. Alpaydm, "Combined 5× 2 cv f test for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 11, no. 8, pp. 1885–1892, 1999.

[46] B. Lin, X. He, C. Zhang, and M. Ji, "Parallel vector field embedding," *The J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2945–2977, 2013.

[47] M. Ji, B. Lin, X. He, D. Cai, and J. Han, "Parallel field ranking," *ACM Trans. Knowl. Discovery Data*, vol. 7, no. 3, p. 15, 2013.

Ziyu Guan received the BS and PhD degrees in computer science from Zhejiang University, China, in 2004 and 2010, respectively. He had worked as a research scientist in the University of California at Santa Barbara from 2010 to 2012. He is currently a full professor in the College of Information and Technology at China's Northwest University. His research interests include machine learning, graph mining and search, expertise modeling and retrieval, and recommender systems.



Lijun Zhang received the BS and PhD degrees in software engineering and computer science from Zhejiang University, China, in 2007 and 2012, respectively. He is currently an associate professor in the Department of Computer Science and Technology, Nanjing University, China. Prior to joining Nanjing University, he was a postdoctoral researcher in the Department of Computer Science and Engineering, Michigan State University. His research interests include machine learning, optimization, information retrieval, and data mining.

Jinye Peng received the MS degree in radio electronics from Northwest University, Xian, China, in 1996 and the PhD degree in signal and information processing from Northwestern Polytechnical University, Xian, China, in 2002. He has published two books and more than 160 scientific papers in journals and conference proceedings. He became a full professor at Northwest University in 2003. He was awarded as "New Century Excellent Talent" by the Ministry of Education of China in 2007. His research interests include image/video analysis and retrieval and face recognition.

Jianping Fan is a full professor in the College of Information and Technology at China's Northwest University. His research interests include statistical machine learning and computer vision.