Deep Multi-View Concept Learning

Cai Xu[†], Ziyu Guan[†*], Wei Zhao[†], Yunfei Niu[†], Quan Wang[‡], Zhiheng Wang[♯]
[†]State Key Lab of ISN, School of Computer Science and Technology, Xidian University
[‡]School of Computer Science and Technology, Xidian University
[♯]College of Computer Science and Technology, Henan Polytechnic University
{cxu_3@stu., zyguan@, ywzhao@mail., yfniu@stu., qwang@}xidian.edu.cn, wzhenry@eyou.com

Abstract

Multi-view data is common in real-world datasets, where different views describe distinct perspectives. To better summarize the consistent and complementary information in multi-view data, researchers have proposed various multi-view representation learning algorithms, typically based on factorization models. However, most previous methods were focused on shallow factorization models which cannot capture the complex hierarchical information. Although a deep multi-view factorization model has been proposed recently, it fails to explicitly discern consistent and complementary information in multi-view data and does not consider conceptual labels. In this work we present a semi-supervised deep multi-view factorization method, named Deep Multi-view Concept Learning (DMCL). DMCL performs nonnegative factorization of the data hierarchically, and tries to capture semantic structures and explicitly model consistent and complementary information in multi-view data at the highest abstraction level. We develop a block coordinate descent algorithm for DMCL. Experiments conducted on image and document datasets show that DMCL performs well and outperforms baseline methods.

1 Introduction

Multi-view data is prevalent in many real-world applications. For instance, the same news can be obtained from various language sources; an image can be described by different low level visual features. These views often represent diverse and complementary information of the same data. Integrating multiple views is helpful to boost the performance of data mining tasks. We are concerned with representation learning by synthesizing multi-view data. In recent years, a lot of multi-view representation learning algorithms were proposed based on different techniques (e.g. matrix factorization [Guan *et al.*, 2015; Deng *et al.*, 2015], transfer learning [Xu and Sun, 2012]).

*Corresponding author

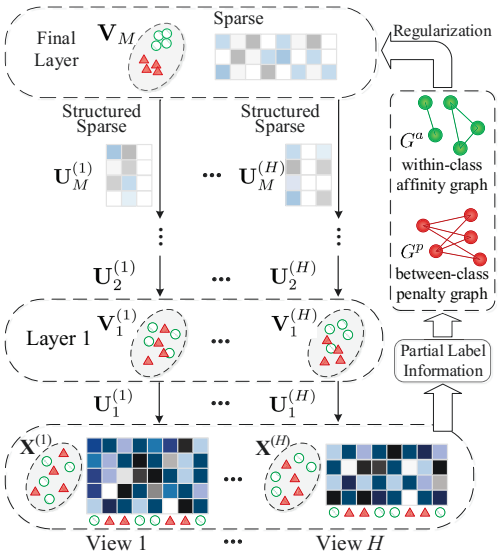


Figure 1: Illustration of DMCL. It factorizes multi-view data iteratively to extract the high-level common encoding $\mathbf{V}_M$. Partial label information is leveraged to learn semantic structures and structured sparseness constraints are used to model consistency and complementarity among different views.

As a particularly useful family of techniques in data analysis, matrix factorization is a successful representation learning technique over a variety of areas, e.g. recommendation [Wang *et al.*, 2016a; 2016b], image clustering [Trigeorgis *et al.*, 2017]. Recently, Nonnegative Matrix Factorization (NMF), a specific form of matrix factorization, has received significant attention in multi-view representation learning [Zong *et al.*, 2017; Guan *et al.*, 2015; Liu *et al.*, 2015] due to its intuitive parts-based interpretation [Lee and Seung, 2001]. Given a data matrix $\mathbf{X} \in \mathbb{R}_+^{D\times N}$ for N items, NMF seeks two nonnegative matrices $\mathbf{U} \in \mathbb{R}_+^{D\times K}$ and $\mathbf{V} \in \mathbb{R}_+^{K\times N}$ such that $\mathbf{X} \approx \mathbf{UV}$. $\mathbf{U}/\mathbf{V}$ is called the basis/encoding matrix. The Multi-view Concept Learning (MCL) algorithm proposed in [Guan *et al.*, 2015] is a typical semi-supervised method which explicitly discerns consistent and complementary information in multi-view data to generate conceptual representations. However, a common draw-

back of the above methods is that they fail to capture complex hierarchical structures of real-world data.

In order to learn a better representation by capturing the hierarchical structures, different deep matrix factorization techniques were proposed recently. Trigeorgis *et al.* [Trigeorgis *et al.*, 2017] proposed the Deep Semi-NMF method for representation learning. It has an interpretation of clustering according to different attributes of a given dataset. Nevertheless, it only deals with the single view case. Zhao *et al.* [Zhao *et al.*, 2017] extended Deep Semi-NMF to the multi-view case. Although it is a multi-view deep factorization method, it neither considers label information of data nor explicitly models consistent and complementary information.

In this paper, we propose a new multi-view deep factorization method, named Deep Multi-view Concept Learning (DMCL). As shown in Figure 1, the deep model factorizes the data matrices ($\mathbf{X}^{(v)}$) iteratively to get the final representation $\mathbf{V}_M$. For each view, each layer is a NMF which takes the representation obtained from the previous layer as its data matrix. The final representation $\mathbf{V}_M$ is shared across different views. We impose graph embedding regularization [Yan *et al.*, 2007] on $\mathbf{V}_M$ to capture data conceptual structures. We also require the basis matrices ($\{\mathbf{U}_M^{(v)}\}$) of the final layer to be sparse in term of columns to explicitly model consistency and complementarity among different views.

The major contribution of this work is a novel semi-supervised deep NMF method for conceptual representation learning from multi-view data. Conceptual features can reflect semantic relationships between data items; they are connected to different views in a flexible fashion, i.e. some conceptual features are described by all the views (consistency), while others may only be associated with some of the views (complementarity). We design the optimization problem to encourage $\mathbf{V}_M$ and ($\{\mathbf{U}_M^{(v)}\}$ to comply with these properties. The second contribution is that we propose a block coordinate decent algorithm to optimize DMCL. We also design a pre-training scheme for DMCL. Thirdly, we empirically evaluate DMCL on two real world datasets and show its superiority over state-of-art baseline methods.

2 Related Work

Our work falls into the area of multi-view representation learning, which is concerned with how to embed inputs from different views of the same set of data items to a new common latent space for better data representation. A recent survey for this area can be found in [Li *et al.*, 2016]. One direction stemmed from Canonical Correlation Analysis (CCA) [Chaudhuri *et al.*, 2009] is based on the principle of maximizing correlations in the common latent space. However, it is nontrivial to extend those methods to deal with multiple views. Another popular idea is to find a shared latent representation across different views. Many methods in this direction were based on (nonnegative) matrix factorization, e.g. [Zong *et al.*, 2017; Liu *et al.*, 2015; Guan *et al.*, 2015]. However, researchers were mainly focused on consistency among different views, while complementarity is rarely explicitly modeled. There were some works that explicitly considered complementarity. Guan *et al.* proposed a NMF-based flexible method where group sparseness constraints were imposed on basis matrices to learn flexible association patterns between encoding dimensions and views [Guan *et al.*, 2015].

Nevertheless, traditional models are intrinsically shallow models and may not well handle intricate natural data. Inspired by deep learning [Bengio and others, 2009], different deep models were proposed recently for multi-view representation learning. Srivastava and Salakhutdinov [Srivastava and Salakhutdinov, 2012] proposed to learn joint representation of images and texts by Deep Boltzmann Machines. Ngiam *et al.* [Ngiam *et al.*, 2011] explored extracting shared representations by training a bimodal deep autoencoder. Deep matrix factorization techniques that factorize complex natural data into multiple levels of factors will also increase representational and modeling power [Sharma *et al.*, 2017; Trigeorgis *et al.*, 2017]. Those methods can be viewed as a decoder network that produces a reconstruction $\mathbf{X} = g(\mathbf{V}_M)$. Compared to deep matrix factorization, deep neural networks are harder to approximate global optima and lack interpretability. A closely related work is [Zhao *et al.*, 2017] where multi-layer matrix factorization is performed for multi-view data clustering. Our DMCL is different from their method in that we not only incorporate label information but also explicitly learn consistency and complementarity among multiple views, trying to capture conceptual features hidden in the data.

3 The Method

Our DMCL is a deep extension of the MCL method. In this section, we review MCL briefly and then present DMCL, together with its optimization algorithm.

3.1 A Brief Review of MCL

We use $\mathbf{X}^{(v)} \in \mathbb{R}_+^{D_v \times N}$ to denote the v-th view of data, where D_v is the dimensionality of the v-th view. The dataset is described by H views: $\{\mathbf{X}^{(v)}\}_{v=1}^H$. The basis matrix $\mathbf{U}^{(v)} \in \mathbb{R}_+^{D_v \times K}$ denotes the linear connection between $\mathbf{X}^{(v)}$ and $\mathbf{V}$, the common encoding matrix. The data matrix of each view is separated into labeled and unlabeled parts: $\mathbf{X}^{(v)} = [\mathbf{X}^{(v),l}\ \mathbf{X}^{(v),u}]$. Correspondingly, the encoding matrix becomes $\mathbf{V} = [\mathbf{V}^l\ \mathbf{V}^u]$. We use N^l/N^u to denote the number of labeled/unlabeled items, respectively. The optimization problem of MCL is formualted as [Guan *et al.*, 2015]

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

The upper bound 1 for V_{kj} is used to guarantee the problem is well lower bounded [Guan *et al.*, 2015]. The first term is the reconstruction criterion. The second term contains the group

sparseness constraints imposed on the basis matrix of each view. For view v, it is defined as

$$\|\mathbf{U}^{(v)}\|_{1,\infty} = \sum_{k=1}^{K} \max_{1\le i\le M} |U_{ik}^{(v)}| \tag{2}$$

It means we encourage some basis vectors of $\mathbf{U}^{(v)}$ to be completely 0, so that the corresponding dimensions in $\mathbf{V}$ are not associated with this view. These dimensions could represent complementary information in multi-view data.

The third term is the graph embedding criterion for regularizing $\mathbf{V}$ where $tr(\cdot)$ denotes matrix trace. $\mathbf{L}^a$ and $\mathbf{L}^p$ represent the graph Laplacian matrices of within-class affinity graph G^a and between-class penalty graph G^p with their weighted adjacency matrices defined as

$$W_{ij}^a = \begin{cases} \frac{1}{N_{c_i}^l} - \frac{1}{N^l}, & \text{if } c_i = c_j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$W_{ij}^p = \begin{cases} \frac{1}{N^l}, & \text{if } c_i \neq c_j \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where c_i denotes the label of item i, $N_{c_i}^l$ is the total number of items with label c_i. The graph embedding term intrinsically forces within-class items to be near while keeps between-class items away from ecah other. With simple algebra transformation, we have $\sum_{i,j} W_{ij}^a \|\mathbf{v}_i^l - \mathbf{v}_j^l\|_2^2 = tr[\mathbf{V}^l\mathbf{L}^a\left(\mathbf{V}^l\right)^T]$ and $\sum_{i,j} W_{ij}^p \|\mathbf{v}_i^l - \mathbf{v}_j^l\|_2^2 = tr[\mathbf{V}^l\mathbf{L}^p\left(\mathbf{V}^l\right)^T]$.

The fourth term is a simple L_1 norm regularizer on $\mathbf{V}$ since an item should not have too many conceptual features.

MCL not only tries to capture semantic structures of the data by semi-supervision, but also models consistency and complementarity among different views by group sparsity constraints. However, the shallow computation (i.e. one-step factorization) in MCL may not be able to well handle complex real world data.

3.2 Deep Multi-view Concept Learning

In order to obtain a more expressive representation, DMCL decomposes each of the data matrices $\{\mathbf{X}^{(v)}\}_{v=1}^H$ iteratively to obtain the high-level representation:

$$\mathbf{X}^{(v)} \approx \mathbf{U}_1^{(v)}\mathbf{U}_2^{(v)}\cdots\mathbf{U}_M^{(v)}\mathbf{V}_M \tag{5}$$

where $\mathbf{U}_1^{(v)} \in \mathbb{R}_+^{D_v\times p_1}$,$\cdots$,$\mathbf{U}_m^{(v)} \in \mathbb{R}_+^{p_{m-1}\times p_m}$,$\cdots$,$\mathbf{U}_M^{(v)} \in \mathbb{R}_+^{p_{M-1}\times p_M}$ denote M basis matrices and $\mathbf{V}_M \in \mathbb{R}_+^{p_M\times N}$ denotes the final common encoding. The optimization problem of DMCL is

$$\begin{aligned} \min_{\{\mathbf{U}_m^{(v)}\},\mathbf{V}_M} \quad & \frac{1}{2}\sum_{v=1}^{H}\left\|\mathbf{X}^{(v)} - \mathbf{U}_1^{(v)}\mathbf{U}_2^{(v)}\cdots\mathbf{U}_M^{(v)}\mathbf{V}_M\right\|_F^2 \\ & + \frac{\beta}{2}\left\{tr\left[\mathbf{V}_M^l\mathbf{L}^a\left(\mathbf{V}_M^l\right)^T\right] - tr\left[\mathbf{V}_M^l\mathbf{L}^p\left(\mathbf{V}_M^l\right)^T\right]\right\} \\ & + \alpha\sum_{v=1}^{H}\left\|\mathbf{U}_M^{(v)}\right\|_{1,\infty} + \gamma\|\mathbf{V}_M\|_{1,1} \\ \text{s.t.} \quad & (U_m^{(v)})_{ik} \ge 0, 1 \ge (V_M)_{kj} \ge 0, \quad \forall i,j,k,v,m. \end{aligned} \tag{6}$$

Here we only apply the graph embedding constraints and the encoding sparseness constraint to $\mathbf{V}_M$, since the intermediate encodings are near low-level features, so they would not well represent high-level conceptual features. The group sparseness constraints used to learn the structures of consistency and complementarity are only imposed on $\{\mathbf{U}_M^{(v)}\}$, since they can only be used where a common encoding ($\mathbf{V}_M$) is reached. Note although the overall factorization is equivalent to a linear operation, as in [Zhao *et al.*, 2017], the multi-layer computation can still help to better represent data items hierarchically by seeking a better local optimum [Zhao *et al.*, 2017].

3.3 Optimization

(6) is not convex in both $\{\mathbf{U}_m^{(v)}\}$ and $\mathbf{V}_M$. Therefore, we can only find its local minima. To improve the quality of the solution and speedup learning, we initialize the model parameters using unsupervised greedy pre-training, similar to layer-wise pre-training in deep learning [Hinton and Salakhutdinov, 2006]. Specifically, for each view v we first decompose $\mathbf{X}^{(v)}$ as $\mathbf{X}^{(v)} = \mathbf{U}_1^{(v)}\mathbf{V}_1^{(v)}$ using NMF. Then, we treat the learned $\mathbf{V}_1^{(v)}$ as the "data matrix" for layer 2 and continue to factorize it iteratively until the final layer. An exception is $\mathbf{V}_M$. For layer M, we obtain a set of encoding matrices $\{\mathbf{V}_M^{(v)}\}$ from the above initialization scheme. However, it is difficult to use them to initialize $\mathbf{V}_M$ since elements in the same position of the encoding vectors for an item may represent different meanings in different views and so they are not comparable. Hence, we choose to initialize $\mathbf{V}_M$ randomly. Preliminary experiments also confirmed its effectiveness.

Afterwards, the variables of (6) are separated into three groups: $\{\mathbf{U}_m^{(v)}\}_{m\neq M}$, $\{\mathbf{U}_M^{(v)}\}_{v=1}^H$ and $\mathbf{V}_M$. (6) is convex in one group when the other two are fixed. Therefore, we solve DMCL by block coordinate descent [Lin, 2007] which each time optimizes one group of variables while keeping the other groups fixed. The procedure is depicted in Algorithm 1. At line 2, we have $\mathbf{V}_0^{(v)} := \mathbf{X}^{(v)}$. Next, we describe the detailed ideas for addressing the three subproblems.

Updating $\mathbf{V}_M$

Since $\mathbf{V}_M$ is randomly initialized, we optimize it firstly. The subproblem for $\mathbf{V}_M$ is:

$$\begin{aligned} \min_{\mathbf{V}_M} \quad & \psi(\mathbf{V}_M) := \frac{1}{2}\sum_{v=1}^{H}\|\mathbf{X}^{(v)} - \widetilde{\mathbf{U}}_M^{(v)}\mathbf{V}_M\|_F^2 + \gamma\|\mathbf{V}_M\|_{1,1} \\ & + \frac{\beta}{2}\left\{tr\left[\mathbf{V}_M^l\mathbf{L}^a(\mathbf{V}_M^l)^T\right] - tr\left[\mathbf{V}_M^l\mathbf{L}^p(\mathbf{V}_M^l)^T\right]\right\} \\ \text{s.t.} \quad & 1 \ge (V_M)_{kj} \ge 0, \quad \forall k,j. \end{aligned} \tag{7}$$

where $\widetilde{\mathbf{U}}_M^{(v)} = \prod_{m=1}^M \mathbf{U}_m^{(v)}$.

We can decompose (7) into two subproblems in which the variables are $\mathbf{V}_M^l$ and $\mathbf{V}_M^u$, labeled part and unlabeled part of $\mathbf{V}_M$, respectively. The update rules can be similarly derived as in [Guan *et al.*, 2015]. Here we only give the equations

Algorithm 1: Optimization of DMCL

Input: $\{\mathbf{X}^{(v)}\}_{v=1}^{H}$; α; β; γ; $\mathbf{L}^a$; $\mathbf{L}^p$; layer sizes $\{p_m\}$
Output: $\{\mathbf{U}_m^{(v)}\}, \forall v, m$; $\mathbf{V}_M$

for m = *1* **to** M, v = *1* **to** H **do**
  $\mathbf{U}_m^{(v)}, \mathbf{V}_m^{(v)} \leftarrow \mathrm{NMF}(\mathbf{V}_{m-1}^{(v)}, p_m)$
end
Randomly initialize $\mathbf{V}_M$
repeat
  Fix other variables, optimize (7) w.r.t $\mathbf{V}_M$.
  Fix other variables, optimize (15) w.r.t $\{\mathbf{U}_m^{(v)}\}_{m\neq M}$.
  Fix other variables, optimize (18) w.r.t $\{\mathbf{U}_M^{(v)}\}_{v=1}^{H}$.
until *convergence or max no. iterations reached*

due to space limitation:

$$(V_M^l)_{kj} \leftarrow \min\left\{1, \frac{-B_{kj}+\sqrt{B_{kj}^2+4A_{kj}C_{kj}}}{2A_{kj}}(V_M^l)_{kj}\right\} \tag{8}$$

$$(V_M^u)_{kj} \leftarrow \min\left\{1, \frac{-(\gamma-Q_{kj}^u)+|\gamma-Q_{kj}^u|}{2(\mathbf{P}\mathbf{v}_j^u)_k}(V_M^u)_{kj}\right\} \tag{9}$$

where $\mathbf{P}$, $\mathbf{Q}^u$, A_{kj}, B_{kj} and C_{kj} are

$$\mathbf{P} = \sum_{v=1}^{H}(\widetilde{\mathbf{U}}_M^{(v)})^T\widetilde{\mathbf{U}}_M^{(v)} \tag{10}$$

$$\mathbf{Q}^u = \sum_{v=1}^{H}(\widetilde{\mathbf{U}}_M^{(v)})^T\mathbf{X}^{(v),u} \tag{11}$$

$$A_{kj} = (\mathbf{P}\mathbf{v}_j^l)_k + \beta((\mathbf{D}^a+\mathbf{W}^p)\bar{\mathbf{v}}_k^l)_j \tag{12}$$

$$B_{kj} = \gamma - Q_{kj}^l \tag{13}$$

$$C_{kj} = \beta((\mathbf{D}^p+\mathbf{W}^a)\bar{\mathbf{v}}_k^l)_j \tag{14}$$

Here $\mathbf{v}_j^l$ and $\bar{\mathbf{v}}_k^l$ denote the j-th column vector and the k-th row vector of $\mathbf{V}_M^l$, respectively. $\mathbf{D}^a$ and $\mathbf{D}^p$ are diagonal matrices with $D_{ii}^a = \sum_{j=1}^{N^l} W_{ij}^a$, $D_{ii}^p = \sum_{j=1}^{N^l} W_{ij}^p$.

Updating $\{\mathbf{U}_m^{(v)}\}_{m\neq M}$

Fixing other variables, we get the subproblem for $\{\mathbf{U}_m^{(v)}\}$

$$\begin{aligned}\min_{\{\mathbf{U}_m^{(v)}\}_{m\neq M}} \quad & \Gamma = \frac{1}{2}\sum_{v=1}^{H}\left\|\mathbf{X}^{(v)} - \mathbf{U}_1^{(v)}\mathbf{U}_2^{(v)}\cdots\mathbf{U}_M^{(v)}\mathbf{V}_M\right\|_F^2 \\ \text{s.t.} \quad & (U_m^{(v)})_{ik} \geq 0, \ \ \forall i,k,v; \ m = 1,\cdots,M-1.\end{aligned} \tag{15}$$

For $m = 1$, the updating is similar as in NMF. Otherwise, the gradient of $\mathbf{U}_m^{(v)}$ is derived as:

$$\frac{\partial\Gamma}{\partial\mathbf{U}_m^{(v)}} = (\xi_m^{(v)})^T\xi_m^{(v)}\mathbf{U}_m^{(v)}\Phi_m^{(v)}(\Phi_m^{(v)})^T - (\xi_m^{(v)})^T\mathbf{X}^{(v)}(\Phi_m^{(v)})^T$$

where $\xi_m^{(v)} = \prod_{i=1}^{m-1}\mathbf{U}_i^{(v)}$ and $\Phi_m^{(v)} = \prod_{i=m+1}^{M}\mathbf{U}_i^{(v)}\mathbf{V}_M$. Thus the additive update for $\mathbf{U}_m^{(v)}$ can be given as:

$$(\mathbf{U}_m^{(v)})_{ik} \leftarrow (\mathbf{U}_m^{(v)})_{ik} - \eta\left(\frac{\partial\Gamma}{\partial\mathbf{U}_m^{(v)}}\right)_{ik} \tag{16}$$

Inspired by [Lee and Seung, 2001], to obtain a multiplicative update rule, η is set as $\frac{(\mathbf{U}_m^{(v)})_{ik}}{\left((\xi_m^{(v)})^T\xi_m^{(v)}\mathbf{U}_m^{(v)}\Phi_m^{(v)}(\Phi_m^{(v)})^T\right)_{ik}}$. Meanwhile, the convergence is guaranteed. Correspondingly, the multiplicative update rule is:

$$\left(\mathbf{U}_m^{(v)}\right)_{ik} \leftarrow \left(\mathbf{U}_m^{(v)}\right)_{ik}\frac{\left((\xi_m^{(v)})^T\mathbf{X}^{(v)}(\Phi_m^{(v)})^T\right)_{ik}}{\left((\xi_m^{(v)})^T\xi_m^{(v)}\mathbf{U}_m^{(v)}\Phi_m^{(v)}(\Phi_m^{(v)})^T\right)_{ik}} \tag{17}$$

Updating $\{\mathbf{U}_M^{(v)}\}_{v=1}^{H}$

When other variables are fixed, the $\mathbf{U}_M^{(v)}$ of different views are independent with each other and their subproblems are identical. For clarity, we just focus on one view and omit the superscript (v) temporally:

$$\begin{aligned}\min_{\mathbf{U}_M} \ \ & \phi(\mathbf{U}_M) := \frac{1}{2}\left\|\mathbf{X}-\widetilde{\mathbf{U}}_{M-1}\mathbf{U}_M\mathbf{V}_M\right\|_F^2 \\ & \qquad + \alpha\|\mathbf{U}_M\|_{1,\infty} \\ \text{s.t.} \ \ & (U_M)_{ik} \geq 0, \ \ \forall i,k.\end{aligned} \tag{18}$$

where $\widetilde{\mathbf{U}}_{M-1} = \prod_{m=1}^{M-1}\mathbf{U}_m$.

$\phi(\mathbf{U}_M)$ is the sum of a differentiable function and a general closed convex function. It can be solved using composite gradient mapping [Nesterov, 2013] which was proposed for minimizing such composite functions. The central idea is to iteratively minimize an auxiliary function $m_L(\mathbf{U}_M)$ and adjust the guess of the Lipschitz constant of the first term of $\phi(\mathbf{U}_M)$ so that $\phi(\mathbf{U}_M)$ decreases by the minimizer of $m_L(\mathbf{U}_M)$. Denote $f(\mathbf{U}_M) = \frac{1}{2}\left\|\mathbf{X}-\widetilde{\mathbf{U}}_{M-1}\mathbf{U}_M\mathbf{V}_M\right\|_F^2$ and $\mathbf{U}_M^t$ as the value of $\mathbf{U}_M$ at the t-th iteration. The auxiliary function is set as

$$\begin{aligned}& m_L(\mathbf{U}_M^t;\mathbf{U}_M) = f(\mathbf{U}_M^t) + \alpha\|\mathbf{U}_M\|_{1,\infty} \\ & + \frac{L}{2}\left\|\mathbf{U}_M-\mathbf{U}_M^t\right\|_F^2 + tr[\nabla f(\mathbf{U}_M^t)^T(\mathbf{U}_M-\mathbf{U}_M^t)]\end{aligned} \tag{19}$$

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

$$\nabla f(\mathbf{U}_M^t) = \widetilde{\mathbf{U}}_{M-1}^T\widetilde{\mathbf{U}}_{M-1}\mathbf{U}_M^t\mathbf{V}_M\mathbf{V}_M{}^T - \widetilde{\mathbf{U}}_{M-1}^T\mathbf{X}\mathbf{V}_M{}^T$$

Then we minimize (19) to get a candidate for $\mathbf{U}_M^{t+1}$:

$$\widehat{\mathbf{U}}_M^{t+1} = \arg\min_{\mathbf{U}_M\geq 0} m_L(\mathbf{U}_M^t;\mathbf{U}_M) \tag{20}$$

We have $\phi(\mathbf{U}_M^t) = m_L(\mathbf{U}_M^t;\mathbf{U}_M^t) \geq m_L(\mathbf{U}_M^t;\widehat{\mathbf{U}}_M^{t+1})$ from (20). Furthermore, it has been proved that for $L \geq L_f$

Algorithm 2: Composite Gradient Mapping

Input: $\eta_u > 1, \eta_d > 1$: scaling parameters for L

begin
  Initialize $L_0 : 0 < L_0 \leq L_f$.
  $t = 0$
  repeat
    repeat
      $L = L_t$
      Optimize (20) to get $\widehat{\mathbf{U}}_M^{t+1}$
      if $\phi(\widehat{\mathbf{U}}_M^{t+1}) > m_L(\mathbf{U}_M^t; \widehat{\mathbf{U}}_M^{t+1})$ **then**
        $L = L\eta_u$
      end
    until $\phi(\widehat{\mathbf{U}}_M^{t+1}) \leq m_L(\mathbf{U}_M^t; \widehat{\mathbf{U}}_M^{t+1})$
    $\mathbf{U}_M^{t+1} = \widehat{\mathbf{U}}_M^{t+1}$
    $L_{t+1} = \max(L_0, L/\eta_d)$
    $t = t + 1$
  until *convergence*
end

we have $m_L(\mathbf{U}_M^t; \widehat{\mathbf{U}}_M^{t+1}) \geq \phi(\widehat{\mathbf{U}}_M^{t+1})$ [Nesterov, 2013], leading to an acceptable $\widehat{\mathbf{U}}_M^{t+1}$. Meanwhile, L is inversely proportional to the step size (i.e. $\|\widehat{\mathbf{U}}_M^{t+1} - \mathbf{U}_M^t\|$), so it cannot be set to too large values. Now the problem is to find a suitable L in each iteration. We starts with an estimate L_0 such that $0 < L_0 \leq L_f$, and in each iteration adjusts L until we get $\phi(\widehat{\mathbf{U}}_M^{t+1}) \leq m_L(\mathbf{U}^t; \widehat{\mathbf{U}}_M^{t+1})$. The algorithm for optimizing $\mathbf{U}_M$ is shown in Algorithm 2.

The remaining problem is how to optimize (20) efficiently. The solution is similar to that in [Guan *et al.*, 2015]. We omit the details due to space limitation.

3.4 Time Complexity

The DMCL model is composed of pre-training and fine-turning stages, so we analyze them separately. For simplify, we set the input feature dimensionalities for all views to D; the dimensionalities of all layers are set to p. We use τ to denote the number of iterations for all iterative procedures.

The computational cost for the pre-training stage is $O(\tau H(DNp+MNp^2))$. For fine-turning, the time complexity consists of three subparts. For optimizing $\{\mathbf{U}_m^{(v)}\}_{m\neq M}$, the cost is $O(\tau HM(Dp^2 + Np^2 + DNp + Mp^3))$. For optimizing $\{\mathbf{U}_M^{(v)}\}_{v=1}^H$, we need to run Algorithm 2. Here we give the result: $O(HD((2(\tau + 1) + \log_2 \frac{L_f}{L_0})P + \tau K^2 + Mp^3 + Dp^2))$. Here τ denotes the number iterations of the outer loop of Algorithm 2. For V_M, the time complexity is $O(\tau(Np + Np^2 + (N^l)^2p) + H(Mp^3 + Dp^2))$. The time cost of DMCL is linear in feature dimension D, item number N and view number H. However, it it not linear in layer number M and layer size p. Therefore, it is essential to set layer number and layer sizes of DMCL properly for high performance and low time consumption.

4 Experiments

We evaluate the performance of DMCL on document and image datasets in terms of classification and clustering metrics. Important statistics are summarized in Table 1 and a brief introduction of the datasets is presented below.

Reuters [Amini *et al.*, 2009]. It consists of 111740 documents written in five languages of 6 categories represented as TFIDF vectors. We utilize documents written in English and Italian as two views. For each categories, we randomly choose 200 documents. Totally 1200 documents are used.

ImageNet [Deng *et al.*, 2009]. It is a well known real-world image database that contains roughly 15 million images organized according to the WordNet hierarchy. We randomly select 50 leaf synsets in the hierarchy as categories and sample 240 images from each one. Three kinds of features are extracted as different views, i.e., 64D HSV histogram, 512D GIST descriptors, and 1000D bag of SIFT visual words.

We compare DMCL with the following baseline algorithms: **Deep NMF (DNMF)** [Song *et al.*, 2015] is a deep matrix factorization method for single view data. We apply it on each view and report the best performance. **Concatenation DNMF (ConcatDNMF)** concatenates feature vectors of different views and then applies DNMF. **Deep Multi-view Semi-NMF (DMSNMF)** [Zhao *et al.*, 2017] is an unsupervised matrix factorization method synthesizing multi-view data to capture a uniform representation. **Multi-view Concept Learning (MCL)** [Guan *et al.*, 2015] is a semi-supervised shallow method for multi-view data.

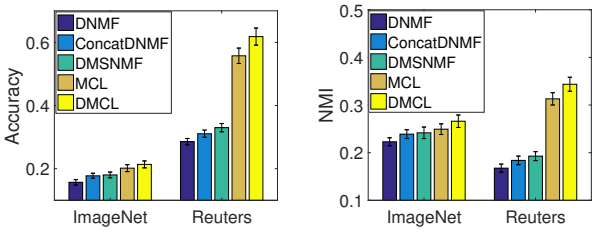
We use the holdout method [Han *et al.*, 2011] for evaluation and tune model parameters by cross-validation on the training set. For each dataset, we randomly split the data items for each category and use 50% for training while the remaining 50% are reserved for test. We use the learned representation of these methods for classification and clustering. Note that the label information of the training set is utilized in semi-supervised methods, i.e., MCL and DMCL. For classification, the training items are fed into the kNN classifier (k=9) and *Accuracy* is calculated using the test set. For clustering, k-means is applied to the test set with k set to the actual number of classes. *Accuracy* and *Normalized Mutual Information(NMI)* are used to evaluate clustering performance [Trigeorgis *et al.*, 2017]. To account for runtime randomness in evaluation, we run each test case 10 times and calculate the averaged performance and standard derivation.

4.1 Results

Table 2 and Figure 2 show the classification and clustering performance of DMCL and baseline methods. First, DNMF is the worst. It is outperformed by all the other methods. This reveals the importance of using multiple views. Second, semi-supervised methods outperforms unsupervised

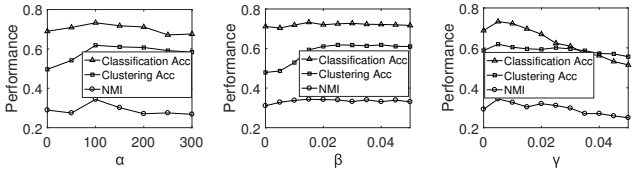
Dataset	Size	# of categories	Dimensionality
Reuters	1200	6	21536/15506
ImageNet	12000	50	64/512/1000

Table 1: Dataset summary.



(a) Accuracy (b) NMI

Figure 2: Clustering performance of different methods. Error bars represent standard deviations.



(a) (b) (c)

Figure 3: Parameter study for DMCL by Accuracy and NMI on the Reuters dataset: (a) varying α when $\beta = 0.015, \gamma = 0.005$; (b) varying β when $\alpha = 100, \gamma = 0.005$; (c) varying γ when $\alpha = 100, \beta = 0.015$.

ones. This is intuitive since by exploiting the partial label information we can build a more discriminative representation. Third, our proposed DMCL consistently outperforms MCL on the two datasets, which indicates that the deep model could generate better representation by hierarchical modeling. We use t-test with significance level 0.05 to test the significance of performance difference. Results show that DMCL significantly outperforms all the baseline methods.

4.2 Analysis

In this subsection, we will analyze DMCL from two perspectives, i.e., parameter setting and convergence analysis.

Parameter analysis. Parameters of DMCL include α, β, γ, and layer number and sizes. Here we explore their impact to performance by cross-validation on the training set and only show representative results due to space limitation. We first focus on the former three parameters. α and γ control the degree of sparseness, while β controls the impact of the graph embedding regularization. We vary one parameter each time and fix the other two to explore its influence. Fig 3 shows the results on Reuters. We find a general pattern: the performance curves first go up and then go down when increasing the parameters. This means the sparseness and graph embedding terms are useful for learning good representations. Based on the results, we set $\alpha = 100$, $\beta = 0.015$ and $\gamma = 0.005$ in other experiments.

Method	ImageNet	Reuters
DNMF	17.66±0.75	59.75±1.65
ConcatDNMF	25.28±0.81	61.25±1.93
DMSNMF	27.39±0.76	64.53±1.72
MCL	30.31±0.62	70.85±1.53
DMCL	32.41±0.67	73.17±1.61

Table 2: Classification performance on different datasets (accuracy±std dev,%).

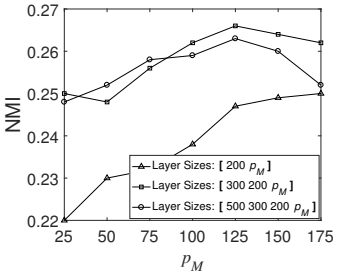
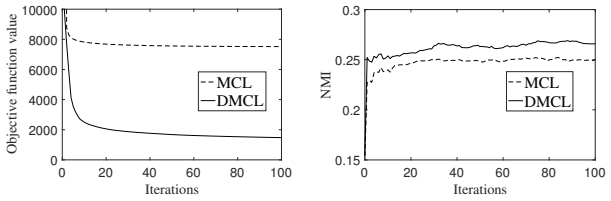


Figure 4: Clustering performance of DMCL with three different layer number settings on the ImageNet dataset.



(a) Objective function value (b) NMI

Figure 5: Convergence analysis of DMCL on ImageNet dataset.

Regarding layer sizes, previous work on multi-view deep factorization [Zhao *et al.*, 2017] has found that p_M, the size of the final layer, usually plays a more important role than the sizes of the other layers. Hence, we vary p_M under different layer numbers and fix the sizes of the other layers empirically. Fig 4 shows the NMI results on ImageNet under 3 layer number settings. The full settings are $\{[200\ p_M],\ [300\ 200\ p_M],\ [500\ 300\ 200\ p_M]\}$. As can be seen, the performance increases with the layer number, which indicates deep factorization really helps to find better representations. Considering both performance and efficiency, we choose 3 layers for all the experiments. The layer sizes are set to $[300\ 200\ 125]$.

Convergence analysis. Fig 5 shows the curves of objective function value and NMI against the number of iterations for DMCL and MCL. We find at the beginning the objective function value drops and the performance increases rapidly. The optimization procedure of DMCL typically converges in around 40 iterations on the ImageNet dataset.

5 Conclusion

In this paper, we developed a Deep Multi-view Concept Learning (DMCL) method to seek a common high-level representation for multi-view partially labeled data. DMCL tries to capture data semantic structures by graph embedding guided by label information. It also tries to learn the consistent and complementary information of multi-view data

by group sparseness constraints. Experimental results on document/image datasets for both classification and clustering tasks confirmed the effectiveness of DMCL compared to competitive multi-view factorization models.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant Nos. 61522206, 61373118, 61672409), the Major Basic Research Project of Shaanxi Province (Grant No. 2017ZDJC-31), the Science and Technology Plan Program in Shaanxi Province of China (Grant No. 2017KJXX-80), the Fundamental Research Funds for the Central Universities, and the Innovation Fund of Xidian University.

References

[Amini *et al.*, 2009] Massih Amini, Nicolas Usunier, and Cyril Goutte. Learning from multiple partially observed views-an application to multilingual text categorization. In *NIPS*, pages 28–36, 2009.

[Bengio and others, 2009] Yoshua Bengio et al. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, 2009.

[Chaudhuri *et al.*, 2009] Kamalika Chaudhuri, Sham M. Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *ICML*, pages 129–136, 2009.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[Deng *et al.*, 2015] Cheng Deng, Zongting Lv, Wei Liu, Junzhou Huang, Dacheng Tao, and Xinbo Gao. Multi-view matrix decomposition: A new scheme for exploring discriminative information. In *IJCAI*, pages 3438–3444, 2015.

[Guan *et al.*, 2015] Ziyu Guan, Lijun Zhang, Jinye Peng, and Jianping Fan. Multi-view concept learning for data representation. *IEEE TKDE*, 27(11):3016–3028, 2015.

[Han *et al.*, 2011] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[Hinton and Salakhutdinov, 2006] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[Lee and Seung, 2001] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2001.

[Li *et al.*, 2016] Yingming Li, Ming Yang, and Zhongfei Zhang. Multi-view representation learning: A survey from shallow methods to deep methods. *arXiv preprint arXiv:1610.01206*, 2016.

[Lin, 2007] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Comput.*, 19(10):2756–2779, 2007.

[Liu *et al.*, 2015] Jing Liu, Yu Jiang, Zechao Li, Zhi-Hua Zhou, and Hanqing Lu. Partially shared latent factor learning with multiview data. *IEEE Trans. Neural Netw. Learn. Syst.*, 26(6):1233–1246, 2015.

[Nesterov, 2013] Yu Nesterov. Gradient methods for minimizing composite functions. *Math. Program.*, 140(1):125–161, 2013.

[Ngiam *et al.*, 2011] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *ICML*, pages 689–696, 2011.

[Sharma *et al.*, 2017] Pulkit Sharma, Vinayak Abrol, Anil Kumar Sao, Pulkit Sharma, Vinayak Abrol, and Anil Kumar Sao. Deep-sparse-representation-based features for speech recognition. *IEEE Trans. Audio Speech Lang. Process.*, 25(11):2162–2175, 2017.

[Song *et al.*, 2015] Hyun Ah Song, Bo-Kyeong Kim, Thanh Luong Xuan, and Soo-Young Lee. Hierarchical feature extraction by multi-layer non-negative matrix factorization network for classification task. *Neurocomputing*, 165:63–74, 2015.

[Srivastava and Salakhutdinov, 2012] Nitish Srivastava and Ruslan R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NIPS*, pages 2222–2230, 2012.

[Trigeorgis *et al.*, 2017] George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, and Björn W Schuller. A deep matrix factorization method for learning attribute representations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(3):417–429, 2017.

[Wang *et al.*, 2016a] Beidou Wang, Martin Ester, Jiajun Bu, Yu Zhu, Ziyu Guan, and Deng Cai. Which to view: Personalized prioritization for broadcast emails. In *WWW*, pages 1181–1190, 2016.

[Wang *et al.*, 2016b] Beidou Wang, Martin Ester, Yikang Liao, Jiajun Bu, Yu Zhu, Ziyu Guan, and Deng Cai. The million domain challenge: Broadcast email prioritization by cross-domain recommendation. In *SIGKDD*, pages 1895–1904, 2016.

[Xu and Sun, 2012] Zhijie Xu and Shiliang Sun. Multi-source transfer learning with multi-view adaboost. In *ICONIP*, pages 332–339, 2012.

[Yan *et al.*, 2007] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):40–51, 2007.

[Zhao *et al.*, 2017] Handong Zhao, Zhengming Ding, and Yun Fu. Multi-view clustering via deep matrix factorization. In *AAAI*, pages 2921–2927, 2017.

[Zong *et al.*, 2017] Linlin Zong, Xianchao Zhang, Long Zhao, Hong Yu, and Qianli Zhao. Multi-view clustering via multi-manifold regularized non-negative matrix factorization. *Neural Netw.*, 88:74–89, 2017.