Multi-View Clustering via Deep Matrix Factorization

Handong Zhao,[†] Zhengming Ding,[†] Yun Fu[†‡]
[†]Department of Electrical and Computer Engineering, Northeastern University, Boston, USA, 02115
[‡]College of Computer and Information Science, Northeastern University, Boston, USA, 02115
{hdzhao,allanding,yunfu}@ece.neu.edu

Abstract

Multi-View Clustering (MVC) has garnered more attention recently since many real-world data are comprised of different representations or views. The key is to explore complementary information to benefit the clustering problem. In this paper, we present a deep matrix factorization framework for MVC, where semi-nonnegative matrix factorization is adopted to learn the hierarchical semantics of multi-view data in a layer-wise fashion. To maximize the mutual information from each view, we enforce the non-negative representation of each view in the final layer to be the same. Furthermore, to respect the intrinsic geometric structure in each view data, graph regularizers are introduced to couple the output representation of deep structures. As a non-trivial contribution, we provide the solution based on alternating minimization strategy, followed by a theoretical proof of convergence. The superior experimental results on three face benchmarks show the effectiveness of the proposed deep matrix factorization model.

Introduction

Traditional clustering aims to identify groups of "similar behavior" in single view data (von Luxburg 2007; Liu et al. 2015; Steinwart 2015; Tao et al. 2016; Liu et al. 2016; Li, Kong, and Fu 2017). As the real-world data are always captured from multiple sources or represented by several distinct feature sets (Cai, Nie, and Huang 2013a; Ding and Fu 2014; Gao et al. 2015; Zhao and Fu 2015; Wang, Ding, and Fu 2016), MVC is intensively studied recently by leveraging the heterogeneous data to achieve the same goal. Different features characterize different information from the data set. For example, an image can be described by different characteristics, e.g., color, texture, shape and so on. These multiple types of features can provide useful information from different views. MVC aims to integrate multiple feature sets together, and uncover the consistent latent information from different views. Extensive research efforts have been made in developing effective MVC methods (Cai, Nie, and Huang 2013a; Gao et al. 2015; Xu, Han, and Nie 2016; Zhao, Liu, and Fu 2016). Along this line, Kumar *et al.* developed co-regularized multi-view spectral clustering to do clustering on different views simultaneously with a co-regularization constraint (Kumar, Rai, and

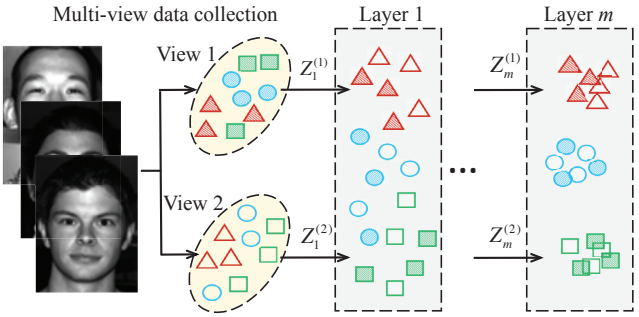

Figure 1: Framework of our proposed method. Same shape denotes the same class. For demonstration purposes, we only show the two-view case, where two deep matrix factorization structures are proposed to capture rich information behind each view in a layer-wise fashion. With the deep structure, samples from the same class but different views gather close to each other to generate more discriminative representation.

III 2011). Gao *et al.* proposed to perform clustering on the subspace representation of each view simultaneously guided by a common cluster structure for the consistence across different views (Gao et al. 2015). A good survey can be found in (Xu, Tao, and Xu 2013).

Recently, lots of research activities on MVC have achieved promising performance based on Non-negative Matrix Factorization (NMF) and its variants, because the non-negativity constraints allow for better interpretability (Guan et al. 2012; Trigeorgis et al. 2014). The general idea is to seek a common latent factor through non-negative matrix factorization among multi-view data (Liu et al. 2013; Zhang et al. 2014; 2015). Semi Non-negative Matrix Factorization (Semi-NMF), as one of the most popular variants of NMF, was proposed to extend NMF by relaxing the factorized basis matrix to be real values. This practice allows Semi-NMF to have a wider application in the real world than NMF. Apart from exploring Semi-NMF in MVC application for the first time, our method has another distinction from the existing NMF-based MVC methods: we adopt a deep structure to conduct Semi-NMF hierarchically as shown in Figure 1. As illustrated, through the deep Semi-NMF structure, we push data samples from the same class closer layer

by layer. We borrow the idea from deep learning (Bengio 2009), thus this practice has such a flavor. Note that the proposed method is different from the existing deep auto-encoder based MVC approaches (Andrew et al. 2013; Wang et al. 2015), though all of us are of deep structure. One major difference is that (Andrew et al. 2013; Wang et al. 2015) are based on Canonical Correlation Analysis (CCA), which is limited to 2-view case, while our method has no such limitation.

To sum up, in this paper we propose a deep MVC algorithm through graph regularized semi-nonnegative matrix factorization. The key is to build a deep structure through semi-nonnegative matrix factorization to seek a common feature representation with more consistent knowledge to facilitate clustering. To the best of our knowledge, this is the first attempt applying semi-nonnegative matrix factorization to MVC in a deep structure. We summarize our major contributions as follows:

- Deep Semi-NMF structure is built to capture the hidden information by leveraging benefits of strong interpretability from Semi-NMF and effective feature learning from deep structure. Through this deep matrix factorization structure, we dissemble unimportant factors layer by layer and generate an effective consensus representation in the final layer for MVC.
- To respect the intrinsic geometric relationship among data samples, we introduce graph regularizers to guide the shared representation learning in each view. This practice makes the consensus representation in the final layer preserve most shared structures across multiple graphs. It can be considered as a fusion scheme to boost the final MVC performance.

Our Method

Overview of Semi-NMF

As a variant of NMF, Ding *et al.* (Ding, Li, and Jordan 2010) extended the application of traditional NMF from nonnegative input to a mix-sign input, while still preserving the strong interpretability at the same time. Its objective function can be expressed as:

$$\min_{Z,H\geq 0} \|X - ZH\|_{\mathrm{F}}^2, \tag{1}$$

where $X \in \mathbb{R}^{d\times n}$ denotes the input data with n samples, each sample is of d dimensional feature. In the discussion on equivalence of semi-NMF and K-means clustering (Ding, Li, and Jordan 2010), $Z \in \mathbb{R}^{d\times K}$ can be considered as the cluster centroid matrix[1], and $H \in \mathbb{R}^{K\times n},\ H \geq 0$ is the "soft" cluster assignment matrix in latent space[2]. Similar to the traditional NMF, the compact representation H uncovers the hidden semantics by simulating the part-based representation in human brain, i.e., psychological and physiological interpretation.

While in reality, natural data may contain different modalities (or factors), e.g., expression, illumination, pose in face datasets (Samaria and Harter 1994; Georghiades, Belhumeur, and Kriegman 2001). Single NMF is not strong enough to eliminate the effect of those undesirable factors and extract the intrinsic class information. To solve this, Trigeorgis *et al.* (Trigeorgis et al. 2014) showed that a deep model based on Semi-NMF has a promising result in data representation. The multi-layer decomposition process can be expressed as

$$\begin{aligned} X &\approx Z_1 H_1^+ \\ X &\approx Z_1 Z_2 H_2^+ \\ &\vdots \\ X &\approx Z_1 \dots Z_m H_m^+ \end{aligned} \tag{2}$$

where Z_i denotes the i-th layer basis matrix, H_i^+ is the i-th layer representation matrix. (Trigeorgis et al. 2014) proved that each hidden representations layer is able to identify the different attributes. Inspired by this work, we propose a MVC method based on deep matrix factorization technique.

The proposed method

In the MVC setting, let us denote $X = \{X^{(1)}, \dots, X^{(v)}, \dots, X^{(V)}\}$ as the data sample set. V represents the number of views. $X^{(v)} \in \mathbb{R}^{d_v\times n}$, where d_v denotes the dimensionality of the v-view data and n is the number of data samples. Then we formulate our model as:

$$\begin{aligned} \min_{\substack{Z_i^{(v)}, H_i^{(v)} \\ H_m, \alpha^{(v)}}} & \sum_{v=1}^{V} (\alpha^{(v)})^{\gamma} \big(\|X^{(v)} - Z_1^{(v)} Z_2^{(v)} \dots Z_m^{(v)} H_m\|_{\mathrm{F}}^2 \\ & + \beta \mathrm{tr}(H_m L^{(v)} H_m^{\mathrm{T}}) \big) \\ \text{s.t. } & H_i^{(v)} \geq 0,\ H_m \geq 0,\ \sum_{v=1}^{V} \alpha^{(v)} = 1,\ \alpha^{(v)} \geq 0, \end{aligned} \tag{3}$$

where $X^{(v)}$ is the given data for v-th view. $Z_i^{(v)},\ i \in \{1, 2, \dots, m\}$ is the i-th layer mapping for view v. m is the number of layers. H_m is the consensus latent representation for all views. $\alpha^{(v)}$ is the weighting coefficient for the v-th view. γ is the parameter to control the weights distribution. $L^{(v)}$ is the graph Laplacian of the graph for view v, where each graph is constructed in k-nearest neighbor (k-NN) fashion. The weight matrix of the graph for view v is $A^{(v)}$ and $L^{(v)} = A^{(v)} - D^{(v)}$, where $D_{ii}^{(v)} = \sum_j A_{ij}^{(v)}$ (He and Niyogi 2004; Ding and Fu 2016).

Remark 1: Due to the homology of multi-view data, the final layer representation $H_m^{(v)}$ for v-th view data should be close to each other. Here, we use the consensus H_m as a constraint to enforce multi-view data to share the same representation after multi-layer factorization.

[1]For a neat presentation, we do not follow the notation style in (Ding, Li, and Jordan 2010), and remove the mix-sign notation "±" on X and Z, which does not affect the rigorousness.

[2]In some literatures (Ding, Li, and Jordan 2010; Zhao et al. 2015), Semi-NMF is also called the soft version of K-means clustering.

Remark 2: Multiple graphs are constructed to constrain the common representation learning so that the geometric structure in each view could be well preserved for the final clustering. Moreover, the novel graph term could fuse the geometric knowledge from multiple views to make the common representation more consistent.

Optimization

To expedite the approximation of the variables in the proposed model, each of the layers is pre-trained to have an initial approximation of variables $Z_i^{(v)}$ and $H_i^{(v)}$ for the i-th layer in v-th view. The effectiveness of pre-training has been proven before (Hinton and Salakhutdinov 2006) on deep autoencoder networks. Similar to (Trigeorgis et al. 2014), we decompose the input data matrix $X^{(v)} \approx Z_1^{(v)} H_1^{(v)}$ to perform the pre-training, where $Z_1^{(v)} \in \mathbb{R}^{d_v \times p_1}$ and $H_1^{(v)} \in \mathbb{R}^{p_1 \times n}$. Then the v-th view feature matrix $H_1^{(v)}$ is decomposed as $H_1^{(v)} \approx Z_2^{(v)} H_2^{(v)}$, where $Z_2^{(v)} \in \mathbb{R}^{p_1 \times p_2}$ and $H_2^{(v)} \in \mathbb{R}^{p_2 \times n}$. p_1 and p_2 are the dimensionalities for layer 1 and layer 2, respectively.[3] Continue to do so until we have pre-trained all layers. Following this, the weights of each layer is fine-tuned by alternating minimizations of the proposed objective function Eq. (3). First, we denote the cost function as $\mathcal{C} = \sum_{v=1}^{V} (\alpha^{(v)})^{\gamma} \big(\|X^{(v)} - Z_1^{(v)} Z_2^{(v)} \dots Z_m^{(v)} H_m\|_{\mathrm{F}}^2 + \beta \mathrm{tr}(H_m L^{(v)} H_m^{\mathrm{T}}) \big)$.

Update rule for weight matrix $\mathbf{Z_i^{(v)}}$. We minimize the objective value with respect to $Z_i^{(v)}$ by fixing the rest of variables in v-th view for the i-th layer. By setting $\partial \mathcal{C} / \partial Z_i^{(v)} = 0$, we give the solutions as

$$\begin{aligned} Z_i^{(v)} &= (\Phi^{\mathrm{T}}\Phi)^{-1} \Phi^{\mathrm{T}} X^{(v)} \tilde{H}_i^{(v)\mathrm{T}} (\tilde{H}_i^{(v)} \tilde{H}_i^{(v)\mathrm{T}})^{-1} \\ Z_i^{(v)} &= \Phi^{\dagger} X^{(v)} \tilde{H}_i^{(v)\dagger}, \end{aligned} \tag{4}$$

where $\Phi = [Z_1^{(v)} \dots Z_{i-1}^{(v)}]$, $\tilde{H}_i^{(v)}$ denotes the reconstruction (or the learned latent feature) of the i-th layer's feature matrix in v-th view, and notation $\dagger$ represents the Moore-Penrose pseudo-inverse.

Update rule for weight matrix $\mathbf{H_i^{(v)}}$ $(\mathbf{i < m})$. Following (Ding, Li, and Jordan 2010), the update rule for $H_i^{(v)}$ $(i < m)$ is formulated as follows:

$$H_i^{(v)} = H_i^{(v)} \odot \sqrt{\frac{[\Phi^{\mathrm{T}} X^{(v)}]^{\mathrm{pos}} + [\Phi^{\mathrm{T}} \Phi H_i^{(v)}]^{\mathrm{neg}}}{[\Phi^{\mathrm{T}} X^{(v)}]^{\mathrm{neg}} + [\Phi^{\mathrm{T}} \Phi H_i^{(v)}]^{\mathrm{pos}}}}, \tag{5}$$

where $[M]^{\mathrm{pos}}$ denotes a matrix that all the negative elements are replaced by 0. Similarly, $[M]^{\mathrm{neg}}$ denotes one that has all the positive elements replaced by 0. That is,

$$\forall k, j \; [M]_{kj}^{\mathrm{pos}} = \frac{|M_{kj}| + M_{kj}}{2}, \; [M]_{kj}^{\mathrm{neg}} = \frac{|M_{kj}| - M_{kj}}{2}. \tag{6}$$

[3]For the ease of presentation, we denote the dimensionalities (layer size) from layer 1 to layer m as $[p_1 \dots p_m]$ in the experiments.

Update rule for weight matrix $\mathbf{H_m}$ (i.e., $\mathbf{H_i^{(v)}}$ $(\mathbf{i = m})$). Since H_m involves the graph term, the updating rule and convergence property have never been investigated before. We give the updating rule first, followed by the proof of its convergence property.

$$H_m = H_m \odot \sqrt{\frac{[\Phi^{\mathrm{T}} X^{(v)}]^{\mathrm{pos}} + [\Phi^{\mathrm{T}} \Phi H_m]^{\mathrm{neg}} + \mathcal{G}_u(H_m, A)}{[\Phi^{\mathrm{T}} X^{(v)}]^{\mathrm{neg}} + [\Phi^{\mathrm{T}} \Phi H_m]^{\mathrm{pos}} + \mathcal{G}_d(H_m, A)}} \tag{7}$$

where $\mathcal{G}_u(H_m, A) = \beta([H_m A^{(v)}]^{\mathrm{pos}} + [H_m D^{(v)}]^{\mathrm{neg}})$ and $\mathcal{G}_d(H_m, A) = \beta([H_m A^{(v)}]^{\mathrm{neg}} + [H_m D^{(v)}]^{\mathrm{pos}})$.

Theorem 1. *The limited solution of the update rule in Eq. (7) satisfies the KKT condition.*

Proof. We introduce the Lagrangian function

$$\begin{aligned} \mathcal{L}(H_m) = \sum_{v=1}^{V} (\alpha^{(v)})^{\gamma} \Big(&\|X^{(v)} - Z_1^{(v)} Z_2^{(v)} \dots Z_m^{(v)} H_m\|_{\mathrm{F}}^2 \\ &+ \beta \mathrm{tr}(H_m L^{(v)} H_m^{\mathrm{T}}) - \eta H_m \Big), \end{aligned} \tag{8}$$

where the Lagrangian multiplier η enforces nonnegative constraints, $H_m \geq 0$. The zero gradient condition gives $\partial \mathcal{L}(H_m) / \partial H_m = 2\Phi^{\mathrm{T}}(\Phi H_m - X^{(v)}) + 2H_m(D^{(v)} - A^{(v)}) - \eta = 0$. From the complementary slackness condition, we obtain

$$\begin{aligned} &\Big(2\Phi^{\mathrm{T}}(\Phi H_m - X^{(v)}) + 2H_m(D^{(v)} - A^{(v)}) \Big)_{kl} (H_m)_{kl} \\ &= \eta_{kl} (H_m)_{kl} = 0. \end{aligned} \tag{9}$$

This is a fixed point equation that the solution must satisfy at convergence.

The limiting solution of Eq. (7) satisfies the fixed point equation. At convergence, $H_m^{(\infty)} = H_m^{(t+1)} = H_m^{(t)} = H_m$, i.e.,

$$(H_m)_{kl} = (H_m)_{kl} \odot \sqrt{\frac{[\Phi^{\mathrm{T}} X^{(v)}]_{kl}^{\mathrm{pos}} + [\Phi^{\mathrm{T}} \Phi H_m]_{kl}^{\mathrm{neg}} + [\mathcal{G}_u(H_m^{(v)}, A)]_{kl}}{[\Phi^{\mathrm{T}} X^{(v)}]^{\mathrm{neg}} + [\Phi^{\mathrm{T}} \Phi H_m]^{\mathrm{pos}} + [\mathcal{G}_d(H_m^{(v)}, A)]_{kl}}}. \tag{10}$$

Note that $\Phi^{\mathrm{T}} X^{(v)} = [\Phi^{\mathrm{T}} X^{(v)}]^{\mathrm{pos}} - [\Phi^{\mathrm{T}} X^{(v)}]^{\mathrm{neg}}$; $\Phi^{\mathrm{T}} \Phi H_m = [\Phi^{\mathrm{T}} \Phi H_m]^{\mathrm{pos}} - [\Phi^{\mathrm{T}} \Phi H_m]^{\mathrm{neg}}$; $H_m D^{(v)} = [H_m D^{(v)}]^{\mathrm{pos}} - [H_m D^{(v)}]^{\mathrm{neg}}$; $H_m A^{(v)} = [H_m A^{(v)}]^{\mathrm{pos}} - [H_m A^{(v)}]^{\mathrm{neg}}$. Thus Eq. (10) reduces to

$$\Big(2\Phi^{\mathrm{T}}(\Phi H_m - X^{(v)}) + 2H_m(D^{(v)} - A^{(v)}) \Big)_{kl} (H_m)_{kl}^2 = 0. \tag{11}$$

Eq. (11) is identical to Eq. (9). Both equations require that at least one of the two factors is equal to zero. The first factors in both equations are identical. For the second factor $(H_m)_{kl}$ or $(H_m^2)_{kl}$, if $(H_m)_{kl} = 0$ then $(H_m^2)_{kl} = 0$, and vice versa. Therefore if Eq. (9) holds, Eq. (11) also holds and vice versa. □

Update rule for weight $\alpha^{(\mathbf{v})}$. Similar to (Cai, Nie, and Huang 2013b), for the ease of representation, let

us denote $\mathcal{R}^{(v)} = \|X^{(v)} - Z_1^{(v)} Z_2^{(v)} \dots Z_m^{(v)} H_m\|_{\mathrm{F}}^2 + \beta \mathrm{tr}(H_m L^{(v)} H_m^{\mathrm{T}})$. The objective in Eq. (3) with respect to $\alpha^{(v)}$ is written as

$$\min_{\alpha^{(v)}} \sum_{v=1}^{V} (\alpha^{(v)})^{\gamma} \mathcal{R}^{(v)}, \ \text{s.t.} \quad \sum_{v=1}^{V} \alpha^{(v)} = 1, \ \alpha^{(v)} \geq 0. \tag{12}$$

The Lagrange function of Eq. (12) is written as

$$\min_{\alpha^{(v)}} \sum_{v=1}^{V} (\alpha^{(v)})^{\gamma} \mathcal{R}^{(v)} - \lambda (\sum_{v=1}^{V} \alpha^{(v)} - 1), \tag{13}$$

where λ is the Lagrange multiplier. By taking the derivative of Eq. (13) with respect to $\alpha(v)$, and setting it to zero, we obtain

$$\alpha^{(v)} = \left(\frac{\lambda}{\gamma \mathcal{R}^{(v)}} \right)^{\frac{1}{\gamma - 1}}. \tag{14}$$

Then we replace $\alpha^{(v)}$ in Eq. (14) into $\sum_{v=1}^{V} \alpha^{(v)} = 1$, and obtain

$$\alpha^{(v)} = \frac{\left(\gamma \mathcal{R}^{(v)}\right)^{\frac{1}{1-\gamma}}}{\sum_{v=1}^{V} \left(\gamma \mathcal{R}^{(v)}\right)^{\frac{1}{1-\gamma}}}. \tag{15}$$

It is interesting to see that with only one parameter γ, we could control the different weights for different views. When γ approaches ∞, we get equal weights. When γ is close to 1, the weight of the view whose $\mathcal{R}^{(v)}$ value is the smallest is assigned to 1, and the others are assigned to 0.

Until now, we have all the update rules done. We repeat the updates iteratively until convergence. The entire algorithm is outlined in **Algorithm 1**. After obtaining the optimized H_m, standard spectral clustering (Ng, Jordan, and Weiss 2001) is performed on the graph built on H_m via k-NN algorithm.

Time complexity

Our deep matrix factorization model is composed of two stages, i.e., pre-training and fine-tuning, so we analyze them separately. To simplify the analysis, we assume the dimensions in all the layers (i.e., layer size) are the same, denoting p. The original feature dimensions for all the views are the same, denoting d. V is the number of views. m is the number of layers.

In pre-training stage, the Semi-NMF process and graph construction are the time consuming parts. The complexity is of order $\mathcal{O}\left(Vmt_p(dnp + np^2 + pd^2 + pn^2 + dn^2)\right)$, where t_p is the number of iterations to achieve convergence in Semi-NMF optimization process. Normally, $p < d$, thus the computational cost is $\mathcal{T}_{pre.} = \mathcal{O}\left(Vmt_p(dnp + pd^2 + dn^2)\right)$ for the pre-training stage. Similarly, in the fine-tuning stage, the time complexity is of order $\mathcal{T}_{fine.} = \mathcal{O}\left(Vmt_f(dnp + pd^2 + pn^2)\right)$, where t_f is the number of iterations in this fine-tuning stage. To sum up, the overall computational cost is $\mathcal{T}_{total} = \mathcal{T}_{pre.} + \mathcal{T}_{fine.}$.

Algorithm 1: Optimization of Problem (3)

Input: Multi-view data $X^{(v)}$, tuning parameters γ, β, layer size p_i, the number of nearest neighbors k.
Initialize:
for *all layers in each view* **do**
  $(Z_i^{(v)}, H_i^{(v)}) \leftarrow \mathrm{SemiNMF}(H_{i-1}^{(v)}, d_i)$
  $\alpha^{(v)} \leftarrow \frac{1}{V}$
  $A^{(v)} \leftarrow k$-NN graph construction on $X^{(v)}$.
end
while *not converged* **do**
  for *all layers in each view* **do**
    $\tilde{H}_i^{(v)} \leftarrow \begin{cases} H_m & \text{if } i = m \\ Z_{i+1}^{(v)} \tilde{H}_{i+1}^{(v)} & \text{otherwise} \end{cases}$
    $\Phi \leftarrow \prod_{\tau=1}^{i-1} Z_\tau$.
    $Z_i \leftarrow \Phi^{\dagger} X^{(v)} \tilde{H}_i^{(v)}$.
    $H_i^{(v)} \leftarrow \begin{cases} \text{Update via Eq. (5)} & \text{if } i < m \\ \text{Update via Eq. (7)} & \text{otherwise} \end{cases}$
    $\alpha^{(v)} \leftarrow$ Update via Eq. (15).
  end
end
Output: Weighted matrices $Z_i^{(v)}$ and feature matrices $H_i^{(v)}$ $(i < m)$ and H_m in the final layer.

Experiments

We choose three face image/video benchmarks in our experiments, as face contains good structural information, which is beneficial to manifesting the strengths of deep NMF structure. A brief introduction of datasets and preprocessing steps is as follows.

Yale consists of 165 images of 15 subjects in raw pixel. Each subject has 11 images, with different conditions, e.g., facial expressions, illuminations, with/without glasses, lighting conditions, etc. **Extended Yale B** consists of 38 subjects of face images. Each subject has 64 faces images under various lighting conditions and poses. In this work, the first 10 subjects, 640 images data are used for experiment. **Notting-Hill** is a well-known video face benchmark (Zhang et al. 2009), which is generated from movie "Notting Hill". There are 5 major casts, including 4660 faces in 76 tracks.

For these datasets, we follow the preprocessing strategy (Cao et al. 2015). Firstly all the images are resized into 48×48 and then three kinds of features are extracted, i.e., intensity, LBP (Ahonen, Hadid, and Pietikäinen 2006) and Gabor (Feichtinger and Strohmer 1998). Specifically, LBP is a 59-dimension histogram over 9×10 pixel patches generated from cropped images. The scale parameter λ in Gabor wavelets is fixed as 4 at four orientations $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ with a cropped image of size 25×30 pixels.

For the comparison baselines, we have the following. (1) **BestSV** performs standard spectral clustering (Ng, Jordan, and Weiss 2001) on the features in each view. We report the best performance. (2) **ConcatFea** concatenates all the features, and then performs standard spectral clustering. (3) **ConcatPCA** concatenates all the features, then projects the

Table 1: Results of 6 different metrics (mean ± standard deviation) on dataset Yale.

Method	NMI	ACC	AR	F-score	Precision	Recall
BestSV	0.654 ± 0.009	0.616 ± 0.030	0.440 ± 0.011	0.475 ± 0.011	0.457 ± 0.011	0.495 ± 0.010
ConcatFea	0.641 ± 0.006	0.544 ± 0.038	0.392 ± 0.009	0.431 ± 0.008	0.415 ± 0.007	0.448 ± 0.008
ConcatPCA	0.665 ± 0.037	0.578 ± 0.038	0.396 ± 0.011	0.434 ± 0.011	0.419 ± 0.012	0.450 ± 0.009
Co-Reg	0.648 ± 0.002	0.564 ± 0.000	0.436 ± 0.002	0.466 ± 0.000	0.455 ± 0.004	0.491 ± 0.003
Co-Train	0.672 ± 0.006	0.630 ± 0.001	0.452 ± 0.010	0.487 ± 0.009	0.470 ± 0.010	0.505 ± 0.007
Min-D	0.645 ± 0.005	0.615 ± 0.043	0.433 ± 0.006	0.470 ± 0.006	0.446 ± 0.005	0.496 ± 0.006
MultiNMF	0.690 ± 0.001	0.673 ± 0.001	0.495 ± 0.001	0.527 ± 0.000	0.512 ± 0.000	0.543 ± 0.000
NaMSC	0.671 ± 0.011	0.636 ± 0.000	0.475 ± 0.004	0.508 ± 0.007	0.492 ± 0.003	0.524 ± 0.004
DiMSC	0.727 ± 0.010	0.709 ± 0.003	0.535 ± 0.001	0.564 ± 0.002	0.543 ± 0.001	0.586 ± 0.003
Ours	$\mathbf{0.782} \pm 0.010$	$\mathbf{0.745} \pm 0.011$	$\mathbf{0.579} \pm 0.002$	$\mathbf{0.601} \pm 0.002$	$\mathbf{0.598} \pm 0.001$	$\mathbf{0.613} \pm 0.002$

Table 2: Results of 6 different metrics (mean ± standard deviation) on dataset Extended YaleB.

Method	NMI	ACC	AR	F-score	Precision	Recall
BestSV	0.360 ± 0.016	0.366 ± 0.059	0.225 ± 0.018	0.303 ± 0.011	0.296 ± 0.010	0.310 ± 0.012
ConcatFea	0.147 ± 0.005	0.224 ± 0.012	0.064 ± 0.003	0.159 ± 0.002	0.155 ± 0.002	0.162 ± 0.002
ConcatPCA	0.152 ± 0.003	0.232 ± 0.005	0.069 ± 0.002	0.161 ± 0.002	0.158 ± 0.001	0.164 ± 0.002
Co-Reg	0.151 ± 0.001	0.224 ± 0.000	0.066 ± 0.001	0.160 ± 0.000	0.157 ± 0.001	0.162 ± 0.000
Co-Train	0.302 ± 0.007	0.186 ± 0.001	0.043 ± 0.001	0.140 ± 0.001	0.137 ± 0.001	0.143 ± 0.002
Min-D	0.186 ± 0.003	0.242 ± 0.018	0.088 ± 0.001	0.181 ± 0.001	0.174 ± 0.001	0.189 ± 0.002
MultiNMF	0.377 ± 0.006	0.428 ± 0.002	0.231 ± 0.001	0.329 ± 0.001	0.298 ± 0.001	0.372 ± 0.002
NaMSC	0.594 ± 0.004	0.581 ± 0.013	0.380 ± 0.002	0.446 ± 0.004	0.411 ± 0.002	0.486 ± 0.001
DiMSC	0.635 ± 0.002	0.615 ± 0.003	0.453 ± 0.000	0.504 ± 0.006	0.481 ± 0.002	0.534 ± 0.001
Ours	$\mathbf{0.649} \pm 0.002$	$\mathbf{0.763} \pm 0.001$	$\mathbf{0.512} \pm 0.002$	$\mathbf{0.564} \pm 0.001$	$\mathbf{0.525} \pm 0.001$	$\mathbf{0.610} \pm 0.001$

Table 3: Results of 6 different metrics (mean ± standard deviation) on dataset Notting-Hill.

Method	NMI	ACC	AR	F-score	Precision	Recall
BestSV	0.723 ± 0.008	0.813 ± 0.000	0.712 ± 0.020	0.775 ± 0.015	0.774 ± 0.018	0.776 ± 0.013
ConcatFea	0.628 ± 0.028	0.673 ± 0.033	0.612 ± 0.041	0.696 ± 0.032	0.699 ± 0.032	0.693 ± 0.031
ConcatPCA	0.632 ± 0.009	0.733 ± 0.008	0.598 ± 0.015	0.685 ± 0.012	0.691 ± 0.010	0.680 ± 0.014
Co-Reg	0.660 ± 0.003	0.758 ± 0.000	0.616 ± 0.004	0.699 ± 0.000	0.705 ± 0.003	0.694 ± 0.003
Co-Train	0.766 ± 0.005	0.689 ± 0.027	0.589 ± 0.035	0.677 ± 0.026	0.688 ± 0.030	0.667 ± 0.023
Min-D	0.707 ± 0.003	0.791 ± 0.000	0.689 ± 0.002	0.758 ± 0.002	0.750 ± 0.002	0.765 ± 0.003
MultiNMF	0.752 ± 0.001	0.831 ± 0.001	0.762 ± 0.000	0.815 ± 0.000	0.804 ± 0.001	0.824 ± 0.001
NaMSC	0.730 ± 0.002	0.752 ± 0.013	0.666 ± 0.004	0.738 ± 0.005	0.746 ± 0.002	0.730 ± 0.011
DiMSC	$\mathbf{0.799} \pm 0.001$	0.843 ± 0.021	0.787 ± 0.001	0.834 ± 0.001	0.822 ± 0.005	0.836 ± 0.009
Ours	0.797 ± 0.005	$\mathbf{0.871} \pm 0.009$	$\mathbf{0.803} \pm 0.002$	$\mathbf{0.847} \pm 0.002$	$\mathbf{0.826} \pm 0.007$	$\mathbf{0.870} \pm 0.001$

original features into a low-dimensional subspace via PCA. Spectral clustering is applied on the projected feature representation. (4) **Co-Reg (SPC)** (Kumar, Rai, and III 2011) co-regularizes the clustering hypotheses to enforce the memberships from different views admit with each other. (5) **Co-Training (SPC)** (Kumar and III 2011) borrows the idea of co-training strategy to alternatively modify the graph structure of each view using other views' information. (6) **Min-D(isagreement)** (de Sa 2005) builds a bipartite graph which derives from the "minimizing-disagreement" idea. (7) **MultiNMF** (Liu et al. 2013) applies NMF to project each view data to the common latent subspace. This method can be roughly considered as one-layer version of our proposed method. (8) **NaMSC** (Cao et al. 2015) firstly applies (Hu et al. 2014) to each view data, then combines the learned representations and feeds to the spectral clustering. (9) **DiMSC** (Cao et al. 2015) investigates the complementary information of representations of multi-view data by introducing a diversity term. This work is also one of the most recent approaches in MVC. We do not make the comparison with deep auto-encoder based methods (Andrew et al. 2013; Wang et al. 2015), because these CCA-based methods cannot fully utilize more than 2 view data, leading to an unfair comparison.

To make a comprehensive evaluation, we use six different evaluation metrics including **normalized mutual information (NMI)**, **accuracy (ACC)**, **adjusted rand index (AR)**, **F-score**, **Precision** and **Recall**. For details about the metrics, readers could refer to (Kumar and III 2011; Cao et al. 2015). For all the metrics, higher value denotes better performance. Different measurements favor different properties, thus a comprehensive view can be acquired from the diverse results. For each experiment, we repeat 10 times and report the mean values along with standard deviations.

Result

Table 1 and Table 2 tabulate the results on datasets Yale and Extended YaleB. Our method outperforms all the other competitors. For the dataset Yale, we raise the performance bar by around 7.57% in NMI, 5.08% in ACC, 8.22% in AR, 6.56% in F-score, 10.13% in Precision and 4.61% in Recall. On average, we improve the state-of-the-art DiMSC by more

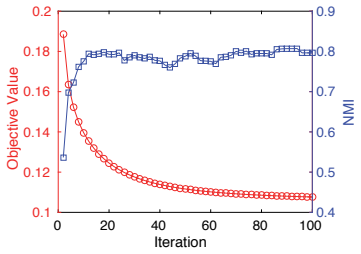

Figure 2: Objective function value (red line) and NMI (blue line) with respect to iteration time on Yale dataset with parameters $\beta = 0.1$, $\gamma = 0.5$ and layer size is [100, 50], respectively.

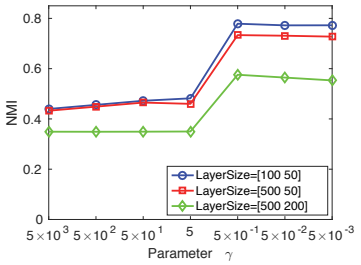

Figure 3: NMI curves w.r.t parameter γ on Yale dataset with three different layer size settings, i.e., {[100 50], [500 50], [500 200]}, and β is set as 0.1.

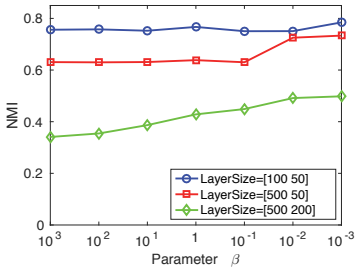

Figure 4: NMI curves w.r.t parameter β on Yale dataset with three different layer size settings, i.e., {[100 50], [500 50], [500 200]}, and γ is set as 0.5.

than 7%. The possible reason why our method improves a lot is that both image data in Yale and Extended YaleB contain multiple factors, i.e., pose, expression, illumination, etc. The existing MVC methods only involve one layer of representation, e.g., one layer factor decomposition in MultiNMF or the practice of self-representation (i.e., coefficient matrix Z in NaMSC and DiMSC (Cao et al. 2015)). However, our proposed approach can extract the meaningful representation layer by layer. Through the deep representation, we eliminate the influence of undesirable factors, and keep the core information (i.e., class/id information) in the final layer.

Table 3 lists the performance on video data Notting-Hill. This dataset is more challenging than the previous two image datasets, since the illumination conditions vary dramatically and the source of lighting is arbitrary. Moreover, there is no fixed expression pattern in the Notting-Hill movie, on the contrary to datasets Yale and Extended YaleB. We observe from the tables that our method reports the superior results in five metrics. The only outlier is NMI, but our performance is slightly worse than DiMSC by only 0.25%. Therefore, we safely draw the conclusion that our proposed method generally achieves better clustering performance in the challenging video dataset Notting-Hill.

Analysis

In this subsection, the robustness and stability of the proposed model is evaluated. The convergence property is firstly studied in terms of objective value and NMI performance. Then the analytical experiments on three key model parameters β, γ, and layer size are conducted.

Convergence analysis. In **Theorem 1**, we theoretically show that the most complex updating for H_m satisfies KKT conditions. To experimentally show the convergence property of the whole model, we compute the objective value of Eq. (3) in each iteration. The corresponding parameters γ, β and layer size are set as 0.5, 0.1 and [100, 50], respectively. The objective value curve is plotted in red in Figure 2. We observe that the objective value decreases steadily, and then gradually meets the convergence after around 100 iterations. The average NMI (in blue) has two stages before converging: from #1 to #14, the NMI increases dramatically; then from #15 to #30, it slightly bumps and reaches the best at around the convergence point. For the sake of safety, the maximum number of iterations is set to 150 for all the experiments.

Parameter analysis. In the proposed method, we have four sets of parameters i.e., balancing parameters β and γ, layer size p_i and the number of nearest neighbors k when constructing k-NN graph. Selecting k in the k-NN graph construction algorithms is an open problem (He and Niyogi 2004). Due to the limited page length, we only include the first three parameter analysis experiments in this paper. However, we find that $k = 5$ usually achieves relatively good results.

Figure 3 shows the influence of NMI result with respect to the parameter γ under three different layer size settings, i.e., {[100 50], [500 50], [500 200]}. Parameter β is set as 0.1. γ is evaluated in the grid of $\{5 \times 10^{-3}, 5 \times 10^{-2}, 5 \times 10^{-1}, 5 \times 10^{0}, 5 \times 10^{1}, 5 \times 10^{2}\}$. Note that to avoid division by 0, γ cannot be set as 1. We observe that the proposed method achieves the best when $\gamma = 0.5$ under different layer size settings. In general, when γ is in the magnitude of $10^{-1}, 10^{-2}, 10^{-3}$, the performance is quite stable. We fix parameter $\gamma = 0.5$ as default in our experiments.

Figure 4 explores the parameter sensitivity of our model in terms of parameter β. Considering the possible amplitude variations of two terms in the objective function Eq. (3), we evaluate β within the following set $\{10^{3}, 10^{2}, 10^{1}, 10^{0}, 10^{-1}, 10^{-2}, 10^{-3}\}$. As can be seen, the average NMI results under three different layer size settings are relatively steady, and slightly better when $\beta = \{10^{-2}, 10^{-3}\}$. In practice, we choose $\beta = 0.01$ as default.

For the layer size analysis, from Figure 3 and Figure 4, we observe that the setting of [100 50] always performs best. Empirically, we find that the last layer dimension usually plays a more important role than other layer size (blue curves are always close to red ones). In Yale dataset, the ground-

truth number of cluster is 10. When the last layer size is set as 200, it might introduce more noise compared with the last layer size set as 50. This is the possible reason why green curves (i.e., layer size is [500 200]) perform worst.

Conclusion

In this paper, we proposed a deep matrix factorization approach for MVC problem. Through the multi-layer Semi-NMF, our method was capable of eliminating the bad influences from diverse modalities, while only keeping the class information in the output layer. With the guidance of multiple graphs, the learned common representation could preserve the geometric structure in each view, especially the common structure information. Extensive experimental results validated the effectiveness of the proposed deep matrix factorization structure, by comparing it with nine baselines.

Acknowledgments

This work is supported in part by the NSF IIS award 1651902, NSF CNS award 1314484, ONR award N00014-12-1-1028, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Young Investigator Award W911NF-14-1-0218.

References

Ahonen, T.; Hadid, A.; and Pietikäinen, M. 2006. Face description with local binary patterns: Application to face recognition. *TPAMI* 28(12):2037–2041.

Andrew, G.; Arora, R.; Bilmes, J.; and Livescu, K. 2013. Deep canonical correlation analysis. In *ICML*, 1247–1255.

Bengio, Y. 2009. Learning deep architectures for AI. *Foundations and Trends in Machine Learning* 2(1):1–127.

Cai, X.; Nie, F.; and Huang, H. 2013a. Multi-view k-means clustering on big data. In *AAAI*, 2598–2604.

Cai, X.; Nie, F.; and Huang, H. 2013b. Multi-view k-means clustering on big data. In *IJCAI*.

Cao, X.; Zhang, C.; Fu, H.; Liu, S.; and Zhang, H. 2015. Diversity-induced multi-view subspace clustering. In *CVPR*, 586–594.

de Sa, V. R. 2005. Spectral clustering with two views. In *ICML*, 20–27.

Ding, Z., and Fu, Y. 2014. Low-rank common subspace for multi-view learning. In *ICDM*, 110–119.

Ding, Z., and Fu, Y. 2016. Robust multi-view subspace learning through dual low-rank decompositions. In *AAAI*, 1181–1187.

Ding, C. H. Q.; Li, T.; and Jordan, M. I. 2010. Convex and semi-nonnegative matrix factorizations. *TPAMI* 32(1):45–55.

Feichtinger, H. G., and Strohmer, T. 1998. *Gabor analysis and algorithms : theory and applications*. Applied and numerical harmonic analysis. Birkhuser.

Gao, H.; Nie, F.; Li, X.; and Huang, H. 2015. Multi-view subspace clustering. In *ICCV*, 4238–4246.

Georghiades, A. S.; Belhumeur, P. N.; and Kriegman, D. J. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *TPAMI* 23(6):643–660.

Guan, N.; Tao, D.; Luo, Z.; and Yuan, B. 2012. Nenmf: an optimal gradient method for nonnegative matrix factorization. *TSP* 60(6):2882–2898.

He, X., and Niyogi, P. 2004. Locality preserving projections. In *NIPS*, 153–160.

Hinton, G. E., and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *Science* 313:504–507.

Hu, H.; Lin, Z.; Feng, J.; and Zhou, J. 2014. Smooth representation clustering. In *CVPR*, 3834–3841.

Kumar, A., and III, H. D. 2011. A co-training approach for multi-view spectral clustering. In *ICML*, 393–400.

Kumar, A.; Rai, P.; and III, H. D. 2011. Co-regularized multi-view spectral clustering. In *NIPS*, 1413–1421.

Li, J.; Kong, Y.; and Fu, Y. 2017. Sparse subspace clustering by learning approximation ℓ_0 codes. In *AAAI*.

Liu, J.; Wang, C.; Gao, J.; and Han, J. 2013. Multi-view clustering via joint nonnegative matrix factorization. In *SDM*, 252–260.

Liu, H.; Liu, T.; Wu, J.; Tao, D.; and Fu, Y. 2015. Spectral ensemble clustering. In *KDD*.

Liu, H.; Shao, M.; Li, S.; and Fu, Y. 2016. Infinite ensemble for image clustering. In *KDD*.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *NIPS*, 849–856.

Samaria, F., and Harter, A. 1994. Parameterisation of a stochastic model for human face identification. In *WACV*, 138–142.

Steinwart, I. 2015. Fully adaptive density-based clustering. *Annals of Statistics* 43(5):2132–2167.

Tao, Z.; Liu, H.; Li, S.; and Fu, Y. 2016. Robust spectral ensemble clustering. In *CIKM*, 367–376.

Trigeorgis, G.; Bousmalis, K.; Zafeiriou, S.; and Schuller, B. W. 2014. A deep semi-nmf model for learning hidden representations. In *ICML*, 1692–1700.

von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and Computing* 17(4):395–416.

Wang, W.; Arora, R.; Livescu, K.; and Bilmes, J. 2015. On deep multi-view representation learning. In *ICML*.

Wang, S.; Ding, Z.; and Fu, Y. 2016. Coupled marginalized auto-encoders for cross-domain multi-view learning. In *IJCAI*, 2125–2131.

Xu, J.; Han, J.; and Nie, F. 2016. Discriminatively embedded k-means for multi-view clustering. In *CVPR*.

Xu, C.; Tao, D.; and Xu, C. 2013. A survey on multi-view learning. *CoRR* abs/1304.5634.

Zhang, Y.; Xu, C.; Lu, H.; and Huang, Y. 2009. Character identification in feature-length films using global face-name matching. *TMM* 11(7):1276–1288.

Zhang, X.; Zhao, L.; Zong, L.; Liu, X.; and Yu, H. 2014. Multi-view clustering via multi-manifold regularized nonnegative matrix factorization. In *ICDM*, 1103–1108.

Zhang, X.; Zong, L.; Liu, X.; and Yu, H. 2015. Constrained nmf-based multi-view clustering on unmapped data. In *AAAI*, 3174–3180.

Zhao, H., and Fu, Y. 2015. Dual-regularized multi-view outlier detection. In *IJCAI*, 4077–4083.

Zhao, H.; Ding, Z.; Shao, M.; and Fu, Y. 2015. Part-level regularized semi-nonnegative coding for semi-supervised learning. In *ICDM*, 1123–1128.

Zhao, H.; Liu, H.; and Fu, Y. 2016. Incomplete multi-modal visual data grouping. In *IJCAI*, 2392–2398.