AE^2-Nets: Autoencoder in Autoencoder Networks

Changqing Zhang[1*], Yeqing Liu[1*], Huazhu Fu[2]
[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]Inception Institute of Artificial Intelligence, Abu Dhabi, UAE
{zhangchangqing; yeqing}@tju.edu.cn; hzfu@ieee.org

Abstract

Learning on data represented with multiple views (e.g., multiple types of descriptors or modalities) is a rapidly growing direction in machine learning and computer vision. Although effectiveness achieved, most existing algorithms usually focus on classification or clustering tasks. Differently, in this paper, we focus on unsupervised representation learning and propose a novel framework termed Autoencoder in Autoencoder Networks (AE^2-Nets), which integrates information from heterogeneous sources into an intact representation by the nested autoencoder framework. The proposed method has the following merits: (1) our model jointly performs view-specific representation learning (with the inner autoencoder networks) and multi-view information encoding (with the outer autoencoder networks) in a unified framework; (2) due to the degradation process from the latent representation to each single view, our model flexibly balances the complementarity and consistence among multiple views. The proposed model is efficiently solved by the alternating direction method (ADM), and demonstrates the effectiveness compared with state-of-the-art algorithms.

1. Introduction

Real-world data are usually described with multiple modalities or multiple types of descriptors that are considered as multiple views. Basically, due to the diversity of sensors or feature extractors, these different views are usually highly heterogeneous. For example, an image may be described with color (*e.g.*, color histogram) and texture descriptors (*e.g.*, SIFT [18], GIST [21], HOG [7]). In social networks, there usually exist both link graph describing relationships between different subjects and subject-specific attributes [31, 28]. In medical image analysis [10], a subject may be associated with different types of medical images used to capture different characteristics of anatomical structures. Accordingly, plenty of approaches have been proposed to jointly exploit multiple types of features [9] or multiple modalities of data [26, 20].

Most existing multi-view learning algorithms focus on classification [13, 4] or clustering [5, 16, 32]. Basically, integrating different views into one comprehensive representation is of vital importance for downstream tasks since unified representation could be easily exploited by on-shelf algorithms. Although it is important, jointly exploring multiple views is a long-standing challenge due to the complex correlations underlying different views. The representative way of learning a common representation is Canonical Correlation Analysis (CCA) [14], which searches for two projections to map two views onto a low-dimensional common subspace where the linear correlation between the two views is maximized. Then the learned representation can be used for subsequent tasks (*e.g.*, classification or clustering). To address more complex correlations beyond linear case, kernelized CCA (KCCA) [1] introduces kernel techniques. Furthermore, Deep Canonical Correlation Analysis (DCCA) [2] proposes learning highly nonlinear mappings with deep neural networks to search for a common space that could maximize the correlations between two views. Beyond CCA-based methods, Partial Least Squares (PLS) regression [25] regresses the samples from one view to another and the flexible multi-view dimensionality co-reduction algorithm (MDcR) [33] maximizes the correlations between different views in kernel space.

Although effectiveness has been achieved on multi-view learning, there are several main problems left for existing algorithms. First, previous algorithms usually project different views onto a common space under the underlying assumption that there exist sufficient correlations between different views. However, in practice, correlation (consistence) and independence (complementarity) are co-existing and it is challenging to automatically balance them. Accordingly, existing algorithms either maximize the correlations [2, 16] for consistence or maximize the independence for complementarity [5]. Second, existing algorithms usually project each view onto a low-dimensional space and then

*Changqing Zhang and Yeqing Liu contributed equally to this work.

combine all of them for subsequent tasks rather than learn a common low-dimensional representation, which makes it a two-step manner in representation learning. Therefore, in this paper, we propose the Autoencoder in Autoencoder Networks (AE^2-Nets), which aims to automatically encode intrinsic information from heterogeneous views into a comprehensive representation and adaptively balance the complementarity and consistence among different views.

The key advantage of the proposed model lies in the joint view-specific encoding and multi-view encoding with a novel nested autoencoder networks. The view-specific representation encoded by the inner-AE networks is responsible for reconstructing the raw input, while the multi-view representation encoded by the outer-AE networks can reconstruct the encoded representation by inner-AE network of each single view. The main contributions of this paper are summarized as follows:

• We propose a novel unsupervised multi-view representation learning framework - Autoencoder in Autoencoder Networks (AE^2-Nets) for heterogeneous data, which can flexibly integrate multiple heterogeneous views into an intact representation.

• The novel nested autoencoder networks could jointly perform view-specific representation learning and multi-view representation learning - the inner autoencoder networks effectively extract information from each single view, while the outer autoencoder networks model the degradation process to encode intrinsic information from each single view into a common intact representation.

• Extensive experimental results verify the effectiveness of the proposed AE^2-Nets on diverse benchmark datasets for both classification and clustering tasks.

The remainder of the paper is organized as follows. Related algorithms, including multi-view learning and multi-view representation learning are briefly reviewed in Section 2. Details of our proposed approach are presented in Section 3. In Section 4, we present experimental results that demonstrate the effectiveness of our model on a variety of real-world datasets. Conclusions are drawn in Section 5.

2. Related Work

Learning based on data with multiple modalities or multiple types of features aims to conduct learning task by jointly utilizing different views to exploit the complementarity, and has attracted intensive attentions recently. For supervised learning, *multimodal metric learning* [34, 35] usually jointly learns multiple metrics for different modalities. Hierarchical Multimodal Metric Learning (HM3L) [35] decomposes the metric of each modality into a product of two matrices: one is modality-specific, and the other is shared by all the modalities. Beyond linear case, Fisher-HSIC Multi-View Metric Learning (FISH-MML) [34] enforces the class separability with Fisher discriminant analysis (FDA) within each view, and maximizes the consistence in kernel space among multiple views by using Hilbert-Schmidt Independence Criteria (HSIC). Under the probabilistic framework, the method [30] learns latent representations and distance metric from multiple modalities with the multi-wing harmonium (MWH) learning. There are also some methods [22, 23] aggregating decisions from multiple classifiers, where each classifier is learned based upon one single modality. Under specific assumptions, theoretical results [11, 6] have advocated the advantages of multi-view integration for subsequent tasks. For clustering, based on spectral clustering, co-regularized [16] and co-training [15] based algorithms enforce clustering hypothesis of different views to be consistent. Recently, the multi-view subspace clustering methods [5, 12] relate different data points in a self-representing manner on the original view and simultaneously constrain these subspace representations of different views to exploit complementary information. There are some multi-view methods focusing on other topics, *e.g.*, dimensionality reduction [33].

Unsupervised multi-view representation learning is a rather challenging problem since there is no class information guiding the learning process. The main stream of methods are CCA-based, which searches for projections to maximize the correlation of two views. Due to the ability of handling nonlinear correlations, the kernel extension of CCA has been widely used for integrating multi-view features or dimensionality reduction. The Deep CCA [2] aims to learn two deep neural networks (DNN) to maximize canonical correlation across two views. Under the deep learning framework, the autoencoder based model [20] learns a compact representation best reconstructing the input. Different from CCA, based on HSIC, a flexible multi-view dimensionality co-reduction method [33] is proposed which explores the correlations within each view independently, and maximizes the dependence among different views with kernel matching jointly. Inspired by deep learning, semi-nonnegative matrix factorization is extended to obtain the hierarchical semantics from multi-view data in a layer-wise manner [36]. The learned representations of all views are enforced to be the same in the final layer.

3. Autoencoder in Autoencoder Networks

In this section, we present the AE^2-Nets for learning the intact representations with a set of multi-view samples $\mathcal{X} = \{\mathbf{X}^{(1)}, ..., \mathbf{X}^{(V)}\}$, where $\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n}$ is the feature matrix of the vth view with V, n and d_v being the number of views, number of samples and dimensionality of feature space for the vth view, respectively.

3.1. Proposed Approach

The key goal of AE^2-Nets (as presented in Fig. 1) is to recover an intact latent space which can well reveal the un-

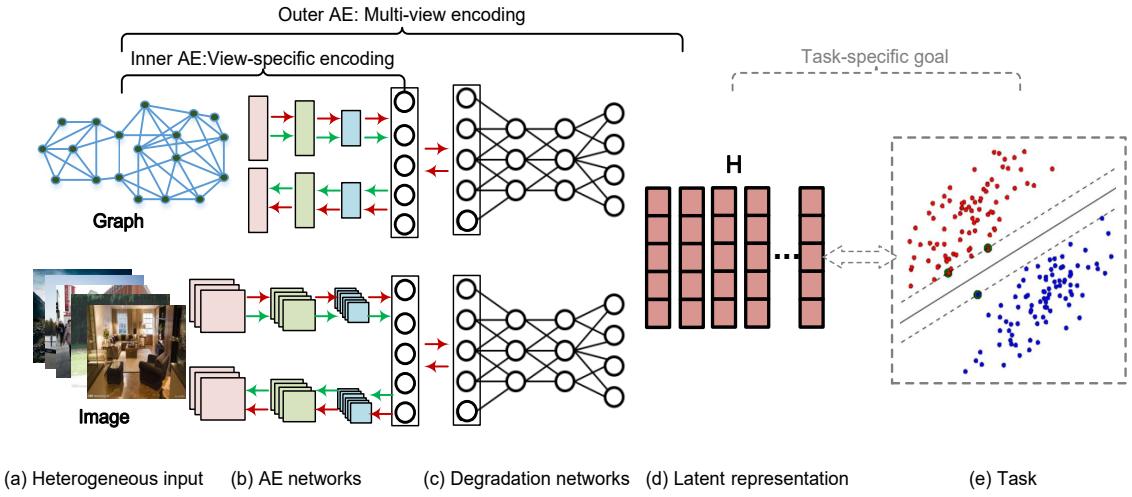


Figure 1: Overview of the Autoencoder in Autoencoder Networks (AE^2-Nets). The key components are the nested autoencoder networks, which are composed of the inner AE networks (shown as the circle with green arrows) for *view-specific encoding* and the outer AE networks (shown as the circle with red arrows) for *multi-view encoding*. *View-specific encoding* automatically extracts features from each view while *multi-view encoding* ensures the intact latent representation can be mapped back to each view with *degradation process*. Accordingly, the intrinsic information from multiple views are encoded into the learned latent intact representation. The learned latent representation could be used for subsequent tasks, and the task-specific goal could flexibly be incorporated into our framework as well (shown in gray dash lines).

derlying structure of data across multiple views. The proposed model jointly learns compact representation for each single view and the intact multi-view representation which can be mapped to reconstruct each single view. Then, the intrinsic information of each view are automatically extracted with the inner-AE networks, and the degradation process involved in the outer-AE networks ensures the intrinsic information from each view are encoded into the latent representation. Note that, due to the common intact representation and associated non-linear networks, more general correlations among different views are addressed.

For the inner networks, the reasons of using AE networks are: (1) since there is no supervised information guiding the learning process, we employ AE networks instead of general neural networks (*e.g.*, for classification) to ensure the intrinsic information to be preserved; (2) for conventional multi-view representation learning models, learning processes are usually based on the pre-extracted features, which is risky due to the high-dimensionality and possible noise involved. The introduced encoding networks could extract intrinsic information to be encoded into the latent multi-view representation instead of the original high-dimensional/noisy features; (3) with variants of AE (*e.g.*, convolutional autoencoder for images), our model has the potential of performing representation learning directly based on raw data.

For simplicity, the inner-AE network for the vth view is denoted as $f(\mathbf{X}^{(v)}; \mathbf{\Theta}_{ae}^{(v)})$, where $\mathbf{\Theta}_{ae}^{(v)} = \{\mathbf{W}_{ae}^{(m,v)}, \mathbf{b}_{ae}^{(m,v)}\}_{m=1}^{M}$ is the parameter set for all layers with $M+1$ being the number of layers of the inner-AE network, *i.e.*, consisting of M layers of nonlinear transformations. Specifically, the first $M/2$ hidden layers encode the input as a new representation, and the last $M/2$ layers decode the representation to reconstruct the input. Let $\mathbf{z}_i^{(0,v)} = \mathbf{x}_i^{(v)} \in \mathbb{R}^{d_v}$ denote an input feature vector, then the output of the mth layer is

$$\begin{aligned} &\mathbf{z}_i^{(m,v)} = a(\mathbf{W}_{ae}^{(m,v)}\mathbf{z}_i^{(m-1,v)} + \mathbf{b}_{ae}^{(m,v)}), \\ &m = 1, 2, \cdots, M, \end{aligned} \tag{1}$$

where $\mathbf{z}_i^{(m,v)} \in \mathbb{R}^{d_{(m,v)}}$ and $d_{(m,v)}$ is the number of nodes at the mth layer for the vth view. $\mathbf{W}_{ae}^{(m,v)} \in \mathbb{R}^{d_{(m,v)} \times d_{(m-1,v)}}$ and $\mathbf{b}_{ae}^{(m,v)} \in \mathbb{R}^{d_{(m,v)}}$ denote the weights and bias associated with the mth layer, respectively. $a(\cdot)$ is a nonlinear activation function. Then, given the feature matrix $\mathbf{X}^{(v)} = [\mathbf{x}_1^{(v)}, \mathbf{x}_2^{(v)}, \cdots, \mathbf{x}_n^{(v)}] \in \mathbb{R}^{d_v \times n}$ for the vth view, the corresponding reconstruct representation is denoted as

$$\mathbf{Z}^{(M,v)} = [\mathbf{z}_1^{(M,v)}, \mathbf{z}_2^{(M,v)}, \cdots, \mathbf{z}_n^{(M,v)}], \tag{2}$$

where $\mathbf{z}_i^{(M,v)}$ is the reconstructed representation for the ith sample in the vth view. To obtain the low-dimensional representation $\mathbf{Z}^{(\frac{M}{2},v)}$, we should minimize the following re-

construction loss

$$\min_{\{\mathbf{\Theta}_{ae}^{(v)}\}_{v=1}^{V}} \frac{1}{2}\sum_{v=1}^{V}\left\|\mathbf{X}^{(v)}-\mathbf{Z}^{(M,v)}\right\|_F^2. \tag{3}$$

After obtaining the low-dimensional view-specific representation $\mathbf{Z}^{(\frac{M}{2},v)}$, we focus on encoding them into one intact common representation, $\mathbf{H}\in\mathbb{R}^{k\times n}$, where k is the dimensionality of the intact space, to preserve intrinsic information from different views. To this end, the degradation networks involved in the outer-AE networks realize the assumption that each single view could be reconstructed from the comprehensive (or intact) common representation. The fully connected neural networks (FCNN) are employed to model the degradation process as shown in Fig. 1(c). Specifically, we map $\mathbf{H}$ onto the view-specific representation $\mathbf{Z}^{(\frac{M}{2},v)}$ with degradation network $g(\mathbf{H};\mathbf{\Theta}_{dg}^{(v)})$, where $\mathbf{\Theta}_{dg}^{(v)}=\{\mathbf{W}_{dg}^{(l,v)},\mathbf{b}_{dg}^{(l,v)}\}_{l=1}^{L}$ with $L+1$ being the number of layers of degradation network. Accordingly, we have $\mathbf{G}^{(0,v)}=\mathbf{H}$ as the input of the degradation networks and $\mathbf{G}^{(l,v)}=[\mathbf{g}_1^{(l,v)},\cdots,\mathbf{g}_n^{(l,v)}]$, with $\mathbf{g}_i^{(l,v)}=a(\mathbf{W}_{dg}^{(l)}\mathbf{g}_i^{(l-1,v)}+\mathbf{b}_{dg}^{(l,v)})$. Then, the objective of degradation networks is defined as

$$\min_{\{\mathbf{\Theta}_{dg}^{(v)}\}_{v=1}^{V}} \frac{1}{2}\sum_{v=1}^{V}\left\|\mathbf{Z}^{(\frac{M}{2},v)}-\mathbf{G}^{(L,v)}\right\|_F^2. \tag{4}$$

In our model, we jointly learn new representation for each view (with inner-AE networks) and seek the intact latent representation (with outer-AE networks) in a unified framework, and then the objective of our AE^2-Nets is induced as

$$\min_{\{\mathbf{\Theta}_{ae}^{(v)},\mathbf{\Theta}_{dg}^{(v)}\}_{v=1}^{V},\mathbf{H}} \frac{1}{2}\sum_{v=1}^{V}\Big(\left\|\mathbf{X}^{(v)}-\mathbf{Z}^{(M,v)}\right\|_F^2 + \lambda\left\|\mathbf{Z}^{(\frac{M}{2},v)}-\mathbf{G}^{(L,v)}\right\|_F^2\Big), \tag{5}$$

where $\lambda>0$ is a tradeoff factor to balance the within-view reconstruction and cross-view reconstruction (from the latent representation to each single view). For all views, $\mathbf{G}^{(L,v)}$s are derived from the common latent representation $\mathbf{H}$. The proposed model automatically learns view-specific representations and nonlinearly encodes them into the multi-view intact representation. It is noteworthy that although the proposed AE^2-Nets is an unsupervised representation learning model, it is easy to extend AE^2-Nets to meet specific tasks (*e.g.*, classification or clustering). Moreover, our model is applicable for the data with more than two views.

3.2. Optimization

There are multiple blocks of variables in our problem, and the objective function of our AE^2-Nets is not jointly convex for all these variables. Therefore, we optimize our objective function by employing Alternating Direction Minimization (ADM) [17] strategy. To adopt the ADM strategy, the optimization is cycled over the following three steps: updating the view-specific auto-encoder networks, updating the degradation networks and updating the latent representation $\mathbf{H}$ by fixing the other blocks of variables. The optimization for each step is as follows:

• **Update View-Specific AE Networks**. To update the view-specific AE network for the vth view, we should minimize the following loss function

$$\mathcal{L}_{ae}^{(v)}(\{\mathbf{\Theta}_{ae}^{(v)}\}_{v=1}^{V}) = \frac{1}{2}\sum_{i=1}^{n}\Big(\left\|\mathbf{x}_i^{(v)}-\mathbf{z}_i^{(M,v)}\right\|^2+\lambda\left\|\mathbf{z}_i^{(\frac{M}{2},v)}-\mathbf{g}_i^{(L,v)}\right\|^2\Big). \tag{6}$$

By applying the chain rule to calculate the gradient of Eq. (6) *w.r.t.* $\mathbf{W}_{ae}^{(m,v)}$ and $\mathbf{b}_{ae}^{(m,v)}$, we have

$$\begin{cases}\dfrac{\partial\mathcal{L}_{ae}^{(v)}}{\partial\mathbf{W}_{ae}^{(m,v)}}=(\mathbf{\Delta}^{(m,v)}+\lambda\mathbf{\Lambda}^{(m,v)})(\mathbf{z}_i^{(m-1,v)})^T,\\ \dfrac{\partial\mathcal{L}_{ae}^{(v)}}{\partial\mathbf{b}_{ae}^{(m,v)}}=\mathbf{\Delta}^{(m,v)}+\lambda\mathbf{\Lambda}^{(m,v)},\end{cases} \tag{7}$$

where $\mathbf{\Delta}^{(m,v)}$ is defined as

$$\mathbf{\Delta}^{(m,v)} = \begin{cases}-(\mathbf{x}_i^{(v)}-\mathbf{z}_i^{(m,v)})\odot a'(\mathbf{y}_i^{(m,v)}),\ m=M,\\ (\mathbf{W}_{ae}^{(m+1,v)})^T\mathbf{\Delta}^{(m+1,v)}\odot a'(\mathbf{y}_i^{(m,v)}),\ \text{otherwise},\end{cases} \tag{8}$$

and $\mathbf{\Lambda}^{(m,v)}$ is given by

$$\mathbf{\Lambda}^{(m,v)} = \begin{cases}(\mathbf{W}_{ae}^{(m+1,v)})^T\mathbf{\Lambda}^{(m+1,v)}\odot a'(\mathbf{y}_i^{(m,v)}),\ m\le\frac{M}{2}-1,\\ (\mathbf{z}_i^{(\frac{m}{2},v)}-\mathbf{g}_i^{(L,v)})\odot a'(\mathbf{y}_i^{(\frac{m}{2},v)}),\ m=\frac{M}{2},\\ \mathbf{0}, m\ge\frac{M}{2}+1.\end{cases} \tag{9}$$

where $a'(\cdot)$ is the derivative of the activation function $a(\cdot)$, $\odot$ denotes the element-wise multiplication, and $\mathbf{y}_i^{(m,v)}=\mathbf{W}_{ae}^{(m,v)}\mathbf{z}_i^{(m-1,v)}+\mathbf{b}_{ae}^{(m,v)}$. Then we can update the parameters $\{\mathbf{W}_{ae}^{(m,v)},\mathbf{b}_{ae}^{(m,v)}\}_{m=1}^{M}$ with gradient descent as

$$\begin{cases}\mathbf{W}_{ae}^{(m,v)}=\mathbf{W}_{ae}^{(m,v)}-\mu\dfrac{\partial\mathcal{L}_{ae}^{(v)}}{\partial\mathbf{W}_{ae}^{(m,v)}},\\ \mathbf{b}_{ae}^{(m,v)}=\mathbf{b}_{ae}^{(m,v)}-\mu\dfrac{\partial\mathcal{L}_{ae}^{(v)}}{\partial\mathbf{b}_{ae}^{(m,v)}},\end{cases} \tag{10}$$

where $\mu>0$ is the learning rate which is usually set to a small positive value, *e.g.*, 0.001.

•**Update Degradation Networks**. Similar to the update strategy for the view-specific AE networks, we can obtain

the gradient of Eq. (4) *w.r.t.* $\mathbf{W}_{dg}^{(l,v)}$ and $\mathbf{b}_{dg}^{(l,v)}$ for the vth view as

$$\frac{\partial \mathcal{L}_{dg}^{(v)}}{\partial \mathbf{W}_{dg}^{(l,v)}} = \boldsymbol{\Upsilon}^{(l,v)}(\mathbf{g}_i^{(l-1,v)})^T, \quad \frac{\partial \mathcal{L}_{dg}^{(v)}}{\partial \mathbf{b}_{dg}^{(l,v)}} = \boldsymbol{\Upsilon}^{(l,v)}, \quad (11)$$

where $\boldsymbol{\Upsilon}^{(l,v)}$ is defined as

$$\boldsymbol{\Upsilon}^{(l,v)} = \begin{cases} -(\mathbf{z}_i^{(\frac{M}{2},v)} - \mathbf{g}_i^{(l,v)}) \odot a'(\mathbf{q}_i^{(l,v)}), \ l = L \\ (\mathbf{W}_{dg}^{(l+1,v)})^T \boldsymbol{\Upsilon}^{(l+1,v)} \odot a'(\mathbf{q}_i^{(l,v)}), \ \text{otherwise} \end{cases} \quad (12)$$

where $\mathbf{q}_i^{(l,v)} = \mathbf{W}_{dg}^{(l,v)}\mathbf{g}_i^{(l-1,v)} + \mathbf{b}_{dg}^{(l,v)}$. Accordingly, we can update the weights and bias with the following rule

$$\begin{cases} \mathbf{W}_{dg}^{(l,v)} = \mathbf{W}_{dg}^{(l,v)} - \mu \dfrac{\partial \mathcal{L}_{dg}^{(v)}}{\partial \mathbf{W}_{dg}^{(l,v)}}, \\ \mathbf{b}_{dg}^{(l,v)} = \mathbf{b}_{dg}^{(l,v)} - \mu \dfrac{\partial \mathcal{L}_{dg}^{(v)}}{\partial \mathbf{b}_{dg}^{(l,v)}}. \end{cases} \quad (13)$$

•**Update Latent Representation $\mathbf{H}$**. To update the intact latent representation $\mathbf{H}$, we follow the similar way as updating $\mathbf{W}_{dg}^{(1,v)}$. That is to say, we should optimize Eq. (4) *w.r.t.* $\mathbf{H}$. Accordingly, we can calculate the gradient as

$$\frac{\partial \mathcal{L}_h}{\partial \mathbf{h}_i} = \sum_{v=1}^{V} \alpha^{(v)}(\mathbf{g}_i^{(L,v)} - \mathbf{z}_i^{(\frac{M}{2},v)}) \odot \prod_{l=1}^{L} a'(\mathbf{q}_i^{(l,v)}) \odot \mathbf{W}_{dg}^{(l,v)}$$
$$\text{with } \mathcal{L}_h = \sum_{v=1}^{V} \frac{\alpha^{(v)}}{2} \left\| \mathbf{z}_i^{(\frac{M}{2},v)} - \mathbf{g}_i^{(L,v)} \right\|^2, \quad (14)$$

where $\alpha^{(v)}$ is a tradeoff factor to control the belief degree for the vth view. In practice, we can set $\alpha^{(1)} = \cdots = \alpha^{(V)}$ when there is no prior about the importance of each view. For clarification, we summarize the optimization procedure in Algorithm 1.

3.3. Connection with CCA/Matrix Factorization

CCA can be interpreted as a generative model [29, 3]. With a latent representation, $\mathbf{h}$, the observations $\mathbf{x}^{(1)} = \mathbf{P}^{(1)}\mathbf{h} + \boldsymbol{\epsilon}^{(1)}$ and $\mathbf{x}^{(2)} = \mathbf{P}^{(2)}\mathbf{h} + \boldsymbol{\epsilon}^{(2)}$, where $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$ are linear mappings, $\boldsymbol{\epsilon}^{(1)}$ and $\boldsymbol{\epsilon}^{(2)}$ are independent Gaussian noise. For our AE^2-Nets, the underlying model is $f(\mathbf{x}^{(v)}; \boldsymbol{\Theta}_{ae}^{(v)}) = g(\mathbf{h}; \boldsymbol{\Theta}_{dg}^{(v)}) + \boldsymbol{\varepsilon}^{(v)}$, where $f(\cdot)$ encodes original features of each view into a compact representation and $g(\cdot)$ degrades the intact representation into each single view. $\boldsymbol{\varepsilon}^{(v)}$ is the error for the vth view. By fixing the features instead of learning by autoencoder networks, and replacing $g(\mathbf{h}; \boldsymbol{\Theta}_{dg}^{(v)})$ with linear projections, our model will be degraded into: $\min_{\{\mathbf{P}^{(v)},\mathbf{H}\}} \sum_{v=1}^{V} \sum_{i=1}^{n} ||\mathbf{x}_i^{(v)} - \mathbf{P}^{(v)}\mathbf{h}_i||^2$. This is similar to the generative model of CCA, and is also equivalent to learning a common representation under the matrix factorization framework.

Algorithm 1: Optimization algorithm of AE^2-Nets

Input: multi-view data $\mathcal{X} = \{\mathbf{X}^{(v)}\}_{v=1}^{V}$, dimensionality k of latent representation $\mathbf{H}$.
Initialize randomly $\{\boldsymbol{\Theta}_{ae}^{(v)}, \boldsymbol{\Theta}_{dg}^{(v)}\}_{v=1}^{V}$ and $\mathbf{H}$.
while *not converged* **do**
  for *each of V views* **do**
    update the parameters of view-specific AE networks with Eq. (10);
  end
  for *each of V views* **do**
    update the parameters of the degradation networks with Eq. (13);
  end
  update $\mathbf{H}$ with Eq. (14);
end
Output: latent representation $\mathbf{H}$.

4. Experiments

In the experiments, we compare the proposed AE^2-Nets with state-of-the-art multi-view representation learning methods on real-world datasets with multiple views, and evaluate the results on both clustering and classification tasks with commonly used evaluation metrics.

4.1. Experimental Settings

Datasets. We conduct the comparisons on the following datasets: **handwritten**[1] contains 2000 images of 10 classes from number 0 to 9. Two different types of descriptors, *i.e.*, pix (240 pixel averages in 2 x 3 windows) and fac (216 profile correlations), are used as two views. **Caltech101-7**[2] contains a subset of images from Caltech101. There are 7 categories selected with 1474 images: faces, motorbikes, dollar-bill, garfield, snoopy, stop-sign, and windsor-chair. The HOG and GIST descriptors are used. **ORL**[3] contains 10 different images for each of 40 distinct subjects. **COIL-20** [4] contains 1440 images of 20 object categories. Each image is normalized to 32×32 with 256 gray levels per pixel. For ORL and COIL-20, intensity of gray level and Gabor descriptors are used. Caltech-UCSD Birds (**CUB**) [5] contains 11788 bird images associated with text descriptions [24] from 200 different categories. We extract 1024-dimensional features based on images with GoogLeNet, and 300-dimensional features based on text.

Compared methods. We compared the proposed AE^2-Nets with the following methods:
(1) **FeatConcate**: This method simply concatenates differ-

[1] https://archive.ics.uci.edu/ml/datasets/Multiple+Features
[2] http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[3] https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
[4] http://www.cs.columbia.edu/CAVE/software/softlib/
[5] http://www.vision.caltech.edu/visipedia/CUB-200.html

Table 1: Performance comparison on clustering task.

Datasets	Methods	ACC	NMI	F_score	RI
handwritten	FeatConcate	76.04 ± 2.28	75.70 ± 1.44	70.96 ± 2.05	93.93 ± 0.42
	CCA [14]	66.43 ± 7.62	69.62 ± 6.06	62.05 ± 7.70	91.83 ± 1.79
	DCCA [2]	66.26 ± 0.16	66.01 ± 0.45	59.05 ± 0.39	91.39 ± 0.06
	DCCAE [27]	69.17 ± 1.02	66.96 ± 0.91	60.50 ± 1.10	91.77 ± 0.21
	MDcR [33]	76.72 ± 2.77	**76.68 ± 0.93**	**71.93 ± 2.22**	**94.11 ± 0.48**
	DMF-MVC [36]	71.86 ± 4.25	73.09 ± 3.23	66.66 ± 4.69	92.85 ± 1.13
	Ours	**81.52 ± 1.62**	71.39 ± 1.50	68.57 ± 1.86	93.68 ± 0.38
Caltech101	FeatConcate	47.23 ± 0.22	57.19 ± 0.61	52.15 ± 0.28	73.45 ± 0.16
	CCA [14]	45.37 ± 0.09	50.53 ± 0.03	52.15 ± 0.19	73.27 ± 0.09
	DCCA [2]	56.71±10.50	57.61 ± 6.78	62.32±12.75	76.34 ± 6.86
	DCCAE [27]	62.11 ± 2.78	**64.38 ± 4.11**	65.43 ± 4.24	79.31 ± 2.06
	MDcR [33]	46.51 ± 0.67	56.43 ± 0.56	51.55 ± 0.56	73.27 ± 0.30
	DMF-MVC [36]	55.75 ± 5.67	45.52 ± 2.28	55.67 ± 5.50	73.43 ± 2.33
	Ours	**66.46 ± 4.55**	60.60 ± 1.93	**73.42 ± 4.91**	**83.14 ± 2.33**
ORL	FeatConcate	61.10 ± 1.51	79.28 ± 0.70	47.03 ± 2.21	97.10 ± 0.25
	CCA [14]	56.98 ± 2.06	76.03 ± 0.79	45.13 ± 1.83	97.32 ± 0.09
	DCCA [2]	59.68 ± 2.04	77.84 ± 0.83	47.72 ± 2.05	97.42 ± 0.13
	DCCAE [27]	59.40 ± 2.20	77.52 ± 0.86	46.71 ± 2.22	97.39 ± 0.14
	MDcR [33]	61.70 ± 2.19	79.45 ± 1.20	48.48 ± 2.59	97.28 ± 0.22
	DMF-MVC [36]	65.38 ± 2.86	82.87 ± 1.26	52.01 ± 3.43	97.29 ± 0.30
	Ours	**68.85 ± 2.11**	**85.73 ± 0.78**	**59.93 ± 1.31**	**97.94 ± 0.11**
COIL20	FeatConcate	67.13 ± 4.09	79.94 ± 1.69	64.81 ± 4.05	96.24 ± 0.60
	CCA [14]	58.68 ± 1.34	70.64 ± 0.47	53.13 ± 0.90	95.18 ± 0.10
	DCCA [2]	63.73 ± 0.78	76.02 ± 0.50	58.76 ± 0.53	95.60 ± 0.06
	DCCAE [27]	62.72 ± 1.40	76.32 ± 0.66	57.56 ± 1.15	95.27 ± 0.30
	MDcR [33]	64.25 ± 2.98	79.44 ± 1.37	63.60 ± 2.57	96.11 ± 0.29
	DMF-MVC [36]	53.92 ± 5.89	72.36 ± 2.11	46.39 ± 4.97	92.56 ± 1.46
	Ours	**73.42 ± 1.90**	**82.55 ± 1.03**	**69.38 ± 1.92**	**96.86 ± 0.22**
CUB	FeatConcate	73.80 ± 0.11	71.49 ± 0.24	61.07 ± 0.18	91.98 ± 0.04
	CCA [14]	45.82 ± 1.58	46.59 ± 0.98	39.93 ± 1.27	87.44 ± 0.31
	DCCA [2]	54.50 ± 0.29	52.53 ± 0.19	45.84 ± 0.31	88.61 ± 0.06
	DCCAE [27]	66.70 ± 1.52	65.76 ± 1.36	58.22 ± 1.18	91.27 ± 0.24
	MDcR [33]	73.68 ± 3.32	74.49 ± 0.75	65.72 ± 1.37	92.75 ± 0.44
	DMF-MVC [36]	37.50 ± 2.45	37.82 ± 2.04	28.95 ± 1.54	85.52 ± 0.26
	Ours	**77.75 ± 1.63**	**78.61 ± 1.62**	**70.96 ± 2.63**	**93.92 ± 0.58**

ent types of features from multiple views.

(2) **CCA**: Canonical Correlation Analysis (CCA) [14] maps multiple types of features onto one common space by finding linear combinations of variables that maximally correlation, and then combines these projected low-dimensional features together.

(3) **DCCA**: Deep Canonical Correlation Analysis (DCCA) [2] extends CCA using deep neural networks, and concatenates projected low-dimensional features of multiple views.

(4) **DCCAE**: Deep Canonically Correlated AutoEncoders (DCCAE) [27] consists of two autoencoders and maximizes the canonical correlation between the learned representations, and then combines these projected low-dimensional features together.

(5) **MDcR**: Multi-view Dimensionality co-Reduction (MDcR) [33] applies the kernel matching to regularize the dependence across multiple views and projects each view onto a low-dimensional space. Then these projected low-dimensional features are concatenated together.

(6) **DMF-MVC**: Deep Semi-NMF for MVC (DMF-MVC) [36] utilizes a deep structure through semi-nonnegative matrix factorization to seek a common feature representation with consistent knowledge for multi-view data.

Evaluation metrics. To comprehensively compare AE^2-Nets with others, we adopt four different metrics to evaluate the clustering quality, *i.e.*, Accuracy, Normalized Mutual Information (NMI), F-score and Rand Index (RI), where different metrics favor different properties of clustering.

Table 2: Performance comparison on classification task.

Datasets	Methods	$G_{80\%}/P_{20\%}$	$G_{70\%}/P_{30\%}$	$G_{50\%}/P_{50\%}$	$G_{20\%}/P_{80\%}$
handwritten	FeatConcate	89.60 ± 1.40	88.97 ± 0.73	88.87 ± 0.44	85.68 ± 0.53
	CCA [14]	93.78 ± 0.82	93.47 ± 0.93	93.28 ± 0.66	91.12 ± 0.74
	DCCA [2]	95.18 ± 0.55	94.62 ± 0.64	94.35 ± 0.46	92.79 ± 0.51
	DCCAE [27]	95.78 ± 0.46	95.10 ± 0.64	94.79 ± 0.58	92.63 ± 0.54
	MDcR [33]	92.33 ± 0.73	91.55 ± 0.39	91.41 ± 0.68	88.11 ± 0.61
	DMF-MVC [36]	94.68 ± 0.71	93.72 ± 0.60	93.33 ± 0.46	88.23 ± 0.57
	Ours	**96.93 ± 0.71**	**96.55 ± 0.66**	**95.88 ± 0.71**	**93.38 ± 0.49**
Caltech101	FeatConcate	87.88 ± 0.67	87.47 ± 0.56	87.17 ± 0.49	87.10 ± 0.45
	CCA [14]	91.10 ± 0.96	90.07 ± 1.03	89.82 ± 0.49	89.08 ± 0.71
	DCCA [2]	92.12 ± 0.58	91.46 ± 0.70	91.30 ± 0.48	90.73 ± 0.38
	DCCAE [27]	91.58 ± 1.02	90.91 ± 0.75	90.54 ± 0.44	89.44 ± 0.43
	MDcR [33]	90.14 ± 0.74	89.45 ± 0.76	88.95 ± 0.41	88.46 ± 0.35
	DMF-MVC [36]	85.51 ± 1.05	84.67 ± 0.82	81.88 ± 0.73	74.19 ± 0.99
	Ours	**93.77 ± 1.35**	**92.98 ± 1.37**	**92.49 ± 0.72**	**91.36 ± 0.69**
ORL	FeatConcate	79.13 ± 2.36	74.58 ± 1.32	68.00 ± 2.23	48.28 ± 2.27
	CCA [14]	77.13 ± 3.96	73.83 ± 4.89	67.95 ± 2.77	49.00 ± 1.84
	DCCA [2]	83.25 ± 2.71	78.92 ± 1.93	71.15 ± 1.86	51.69 ± 1.75
	DCCAE [27]	81.62 ± 2.95	80.00 ± 1.47	72.80 ± 2.04	51.25 ± 1.90
	MDcR [33]	92.00 ± 1.58	90.83 ± 2.08	83.35 ± 1.08	57.38 ± 2.08
	DMF-MVC [36]	93.13 ± 1.21	91.75 ± 1.64	85.45 ± 1.85	56.44 ± 2.50
	Ours	**97.88 ± 1.19**	**96.00 ± 2.18**	**92.20 ± 1.18**	**70.16 ± 2.54**
COIL20	FeatConcate	78.50 ± 2.30	76.42 ± 2.33	67.05 ± 2.33	48.69 ± 2.08
	CCA [14]	90.50 ± 1.46	88.64 ± 0.95	86.86 ± 0.76	78.94 ± 0.87
	DCCA [2]	90.96 ± 1.24	90.48 ± 1.56	88.65 ± 0.84	83.35 ± 0.60
	DCCAE [27]	92.54 ± 0.70	91.88 ± 1.44	90.35 ± 0.58	84.11 ± 1.10
	MDcR [33]	91.11 ± 0.80	90.29 ± 1.05	87.63 ± 1.12	79.46 ± 1.39
	DMF-MVC [36]	95.25 ± 1.06	94.76 ± 0.77	92.07 ± 0.61	82.96 ± 1.03
	Ours	**96.11 ± 1.10**	**95.55 ± 0.87**	**93.25 ± 0.73**	**88.85 ± 0.72**
CUB	FeatConcate	82.50 ± 3.04	81.50 ± 3.13	80.80 ± 1.41	78.33 ± 0.99
	CCA [14]	63.92 ± 3.14	61.39 ± 2.56	59.07 ± 2.32	53.06 ± 2.12
	DCCA [2]	65.67 ± 2.85	64.83 ± 1.83	62.37 ± 1.58	58.44 ± 2.92
	DCCAE [27]	77.00 ± 2.94	74.56 ± 2.74	72.60 ± 2.52	67.35 ± 3.84
	MDcR [33]	83.08 ± 3.43	82.44 ± 3.08	81.53 ± 1.67	78.58 ± 1.65
	DMF-MVC [36]	60.08 ± 2.79	58.56 ± 2.84	55.30 ± 1.90	49.60 ± 1.38
	Ours	**85.83 ± 2.94**	**84.00 ± 1.41**	**82.67 ± 1.41**	**80.17 ± 1.83**

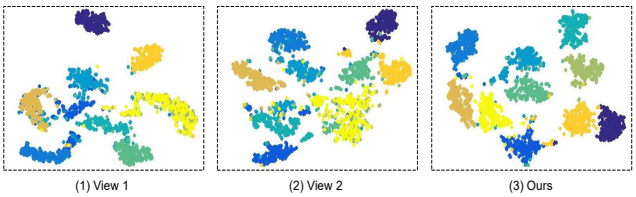
There are different definitions for accuracy for evaluating clustering, and the accuracy used in our experiments is defined as follows: given a sample $\mathbf{x}_i$, its cluster label and class label (ground-truth) are denoted by r_i and s_i, respectively, then we have

$$ACC = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n}, \tag{15}$$

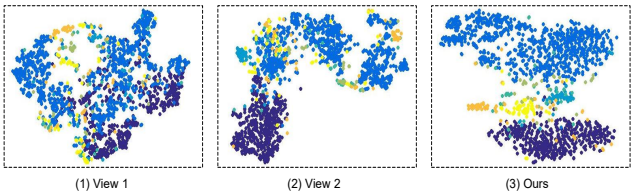
where $\delta(x, y) = 1$ when $x = y$, otherwise $\delta(x, y) = 0$. $map(r_i)$ is the permutation map function, which maps the cluster labels into class labels and the best map can be obtained by Kuhn-Munkres algorithm. We employ the standard classification accuracy and conduct experiments with different partitions of gallery and probe sets. For each of these metrics, a higher value indicates a better clustering performance.

After obtaining the learned representation based on multiple views, we evaluate the learned representation of each method on clustering and classification tasks. For clustering, we employ k-means algorithm, while for classification, k-nearest neighbours (kNN) algorithm is used. The reason for using k-means and kNN lies in the fact that these two algorithms are both simple and can be used based on Euclidean distance to reflect the quality of representation. For all the compared methods, we tune all the parameters to the best performance.

In our model, the fully connected layer with $tanh(\cdot)$ being the activation function is employed for the inner-AE networks and degradation networks, where the numbers of layers for them are empirically set as 5 and 3. We use ℓ_2-norm as regularization for parameters on all network-

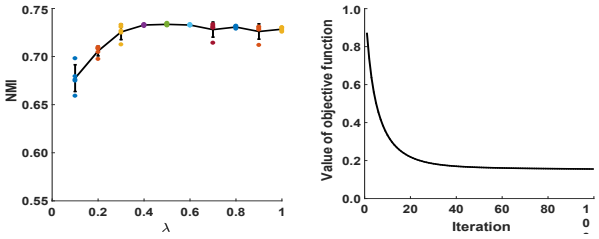


(a) handwritten



(b) Caltech101

Figure 2: Visualization of original features for each single view and the latent representation with t-SNE [19].



(a) (b)

Figure 3: Parameter tuning (a) and convergence curve (b).

s and the weight decay is empirically set to 0.0001. We select the dimensionality of latent representation $\mathbf{H}$ from $\{50, 100, 150, 200, 250, 300\}$ and tune the tradeoff parameter λ from $\{0.1, 0.2, \cdots, 1.0\}$. For simplicity, we set $\alpha_1 = \cdots = \alpha_V = \alpha = 1$ on all datasets. Due to randomness involved, we run all algorithms 30 times and report the mean performances and standard deviations in terms of different metrics.

For clustering, the detailed results of different methods are shown in Table 1. Obviously, our algorithm basically outperforms all the other methods on all datasets in terms of ACC. Since CCA only seeks linear projections, it generally performs rather unpromising. As expected, benefitting from nonlinearity, DCCA and DCCAE perform much better than CCA, which also demonstrates the rationality of our algorithm to model complex correlations based on neural networks instead of linear way. Moreover, although DCCAE and MDcR perform favorably on Caltech101 and handwritten, respectively, it is not promising on other datasets.

For classification, we divide data into different proportions of training and test sets, denoted as $G_{train\_ratio}/P_{test\_ratio}$, where G and P indicate "gallery set" and "probe set", respectively. Table 2 shows the comparison results for each $G_{train\_ratio}/P_{test\_ratio}$. According to Table 2, the accuracy obtained from our AE^2-Nets is more promising than those of comparisons on different partitions. It is observed that CCA-based methods do not always outperform FeatConcate. One possible reason is that overemphasizing the correlation (consistence) may harm the complementarity across different views. The superior performance further validates the advantages of AE^2-Nets.

To further investigate the improvement, we visualize original features of each single view and our learned intact representation with t-SNE [19]. As shown in Fig. 2, the clustering structure is better reflected by the learned latent representation.

Parameter tuning and convergence. The hyperparameter λ is essential to control the fusion of multiple views. As shown in Fig. 3(a), we present the parameter tuning on the handwritten dataset and show the clustering performance of our algorithm with different values for hyperparameter λ. For each value, we repeat 5 times and plot the means and standard deviations in terms of NMI. It is observed that the promising performance could be expected when the value of λ is within a wide range. To demonstrate the convergence of our optimization algorithm, we conduct the convergence experiment as shown in Fig. 3(b). Typically, the objective value decreases fast in the beginning of iterations and our optimization algorithm converges within 100 iterations on these datasets in practice.

5. Conclusion

In this paper, we have presented an unsupervised representation learning model for heterogeneous data. Unlike existing multi-view representation learning models mapping different views onto a common space, the proposed model AE^2-Nets jointly learns the representation of each view and encodes them into an intact latent representation with a novel nested autoencoder framework. In this way, our method can flexibly encode intrinsic information from each view. Experimental results of AE^2-Nets outperform the compared state-of-the-art methods on real-world datasets. For future directions, we will consider extending the current AE^2-Nets for end-to-end representation learning. For example, we can design convolutional AE neural networks for images or graphs [8] for the inner-AE networks to automatically extract features for real-world heterogeneous data.

Acknowledgment

This work was partly supported by National Natural Science Foundation of China (61602337, 61732011, 61702358). Corresponding Author: Changqing Zhang.

References

[1] S. Akaho. A kernel method for canonical correlation analysis. *arXiv preprint cs/0609071*, 2006.

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

[3] F. R. Bach and M. I. Jordan. A probabilistic interpretation of canonical correlation analysis. 2005.

[4] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*, pages 3594–3601, 2010.

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

[6] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. Multi-view clustering via canonical correlation analysis. In *ICML*, pages 129–136, 2009.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, 2005.

[8] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, pages 3844–3852, 2016.

[9] P. Dhillon, D. P. Foster, and L. H. Ungar. Multi-view learning of word embeddings via cca. In *NIPS*, pages 199–207, 2011.

[10] J. S. Duncan and N. Ayache. Medical image analysis: Progress over two decades and the challenges ahead. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):85–106, 2000.

[11] D. P. Foster, S. M. Kakade, and T. Zhang. Multi-view dimensionality reduction via canonical correlation analysis. *Tech Report. Rutgers University*, 2010.

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

[13] K. R. Gray, P. Aljabar, R. A. Heckemann, A. Hammers, D. Rueckert, A. D. N. Initiative, et al. Random forest-based similarity measures for multi-modal classification of alzheimer's disease. *NeuroImage*, 65:167–175, 2013.

[14] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[15] A. Kumar and H. Daumé. A co-training approach for multi-view spectral clustering. In *ICML*, pages 393–400, 2011.

[16] A. Kumar, P. Rai, and H. Daume. Co-regularized multi-view spectral clustering. In *NIPS*, pages 1413–1421, 2011.

[17] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low-rank representation. In *NIPS*, pages 612–620, 2011.

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[19] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[20] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *ICML*, pages 689–696, 2011.

[21] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[22] N. C. Oza and K. Tumer. Classifier ensembles: Select real-world applications. *Information Fusion*, 9(1):4–20, 2008.

[23] Y. Peng, X. Zhou, D. Z. Wang, I. Patwa, D. Gong, and C. Fang. Multimodal ensemble fusion for disambiguation and retrieval. *IEEE MultiMedia*, 2016.

[24] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. pages 49–58, 2016.

[25] A. Sharma and D. W. Jacobs. Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. In *CVPR*, 2011.

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

[27] W. Wang, R. Arora, K. Livescu, and J. Bilmes. On deep multi-view representation learning. pages 1083–1092, 2015.

[28] X. Wang, D. Jin, X. Cao, L. Yang, and W. Zhang. Semantic community identification in large attribute networks. In *AAAI*, pages 265–271, 2016.

[29] M. White, X. Zhang, D. Schuurmans, and Y.-l. Yu. Convex multi-view subspace learning. In *NIPS*, pages 1673–1681, 2012.

[30] P. Xie and E. P. Xing. Multi-modal distance metric learning. In *IJCAI*, pages 1806–1812. Citeseer, 2013.

[31] J. Yang, J. McAuley, and J. Leskovec. Community detection in networks with node attributes. In *ICDM*, pages 1151–1156, 2013.

[32] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, and D. Xu. Generalized latent multi-view subspace clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[33] C. Zhang, H. Fu, Q. Hu, P. Zhu, and X. Cao. Flexible multi-view dimensionality co-reduction. *IEEE Transactions on Image Processing*, 26(2):648–659, 2017.

[34] C. Zhang, Y. Liu, Y. Liu, Q. Hu, X. Liu, and P. Zhu. Fish-mml: Fisher-hsic multi-view metric learning. In *IJCAI*, pages 3054–3060, 2018.

[35] H. Zhang, V. M. Patel, and R. Chellappa. Hierarchical multimodal metric learning for multimodal classification. In *CVPR*, pages 3057–3065, 2017.

[36] H. Zhao, Z. Ding, and Y. Fu. Multi-view clustering via deep matrix factorization. In *AAAI*, pages 2921–2927, 2017.