Shared Generative Latent Representation Learning for Multi-View Clustering

Ming Yin,[*1] **Weitian Huang,**[1] **Junbin Gao**[2]
[1]School of Automation, Guangdong University of Technology, Guangzhou 510006, China.
[2]The University of Sydney Business School, The University of Sydney, Camperdown, NSW 2006, Australia.
yiming@gdut.edu.cn, whytin@yeah.net, junbin.gao@sydney.edu.au.

Abstract

Clustering multi-view data has been a fundamental research topic in the computer vision community. It has been shown that a better accuracy can be achieved by integrating information of all the views than just using one view individually. However, the existing methods often struggle with the issues of dealing with the large-scale datasets and the poor performance in reconstructing samples. This paper proposes a novel multi-view clustering method by learning a shared generative latent representation that obeys a mixture of Gaussian distributions. The motivation is based on the fact that the multi-view data share a common latent embedding despite the diversity among the various views. Specifically, benefitting from the success of the deep generative learning, the proposed model can not only extract the nonlinear features from the views, but render a powerful ability in capturing the correlations among all the views. The extensive experimental results on several datasets with different scales demonstrate that the proposed method outperforms the state-of-the-art methods under a range of performance criteria.

Introduction

Image clustering is one of the fundamental research topics, which has been widely studied in computer vision and machine learning (Caron et al. 2018; Chang et al. 2017). As well, as a class of unsupervised learning methods, clustering has attracted significant attention from various applications. With the advance of information technology, in many real-world scenarios, many heterogeneous visual features, such as HOG (Dalal and Triggs 2005), SIFT (Deng et al. 2009) and LBP (Ojala, Pietikainen, and Maenpaa 2002), can be readily acquired and form a new type data, i.e., multi-view data. These features are collected from different domains or generated from various sensors. Therefore, to efficiently capture the consistency and complementary information among different views, multi-view clustering has gained considerable attention in the recent years (Sun 2013; Xu, Tao, and Xu 2013). In essence, multi-view clustering seeks to partition data points based on multiple representations by assuming that the same cluster structure is shared across all the views (Gao et al. 2015; Wang, Nie, and Huang 2013; Yin et al. 2019). It is crucial for learning algorithm to incorporate the heterogeneous view information to enhance its accuracy and robustness. As well, the effectiveness has been empirically validated under different multi-view scenarios.

In general, multi-view clustering can be roughly separated into two classes, i.e., similarity-based and feature-based. The former aims to construct an affinity matrix whose elements define the similarity between each pair of samples. In light of this, multi-view subspace clustering is one of the most famous similarity-based methods, which purses a latent subspace shared by multiple views, assuming that each view is built from a common subspace (Chaudhuri et al. 2009; Gao et al. 2015; Yin et al. 2019; Zhang et al. 2015). However, these methods often suffer scalability issue due to super-quadratic running time for computing spectra (Jiang et al. 2017). While for the feature-based methods, they seek to partition the samples into K clusters so as to minimize the within-cluster sum of squared errors, such as multi-view K-means clustering (Cai, Nie, and Huang 2013; Xu et al. 2017). It is clear that the selection of feature space is vital as the clustering with Euclidean distance on raw pixels is somehow ineffective.

Inspired by the recent amazing success of deep learning in feature learning (Hinton and Salakhutdinov 2006), a surge of multi-view learning based on deep neural networks (DNN) are proposed (Ngiam et al. 2011; Wang et al. 2015; Xu et al. 2018). First, Ngiam *et al.* (Ngiam et al. 2011) explored extracting shared representations by training a bimodal deep autoencoders. Next, by extending canonical correlation analysis (CCA), Wang *et al.* (Wang et al. 2015) proposed a novel deep canonically correlation autoencoders (DCCAE), which introduces an autoencoders regularization term into deep CCA. However, unfortunately the aforementioned can only be feasible to the two-view case, failing to handle the multi-view one. To explicitly summarize the consensus and complementary information in multi-view data, a Deep Multi-view Concept learning (DMCL) (Xu et al. 2018) is presented by performing non-negative factorization on every view hierarchically.

Though these methods perform well in multi-view clus-

[*]The Project was supported in part by NSF China (No. 61876042).

tering, the generative process of multi-view data cannot be modeled such that they can be used to generate samples accordingly. To this end, benefiting from the success of approximate Bayesian inference, the variational autoencoders (VAE) have been the most popular algorithm under the framework that combines differentiable models with variational inference (Kingma and Welling 2014; Pu et al. 2016). By modeling the data generative procedure with a Gaussian Mixture Model (GMM) model and a neural network, Jiang *et al.* (Jiang et al. 2017) proposed a novel unsupervised generative clustering approach within the framework of VAE, namely Variational Deep Embedding (VaDE). Although it has shown great advantages in clustering, it is not able to be applied *directly* to multi-view learning.

Targeting for classification and information retrieval, Srivastava *et al.* (Srivastava and Salakhutdinov 2014) presented a deep Boltzmann machine for learning a generative model of multi-view data. Until recently there was no successful multi-view extension to clustering yet. The main obstacle is how to efficiently exploit the shared generative latent representation across the views in *unsupervised* way. To tackle this issue, in this paper, we propose a novel multi-view clustering by learning a shared generative latent representation that obeys a mixture of Gaussian distributions, namely Deep Multi-View Clustering via Variational Autoencoders (DMVCVAE). In particular, our motivation is based on the fact that the multi-view data share a common latent embedding despite the diversity among the views. Meanwhile, the proposed model benefits from the success of the deep generative learning, which can capture the data distribution by neural networks.

In summary, our contributions are as follows.

- We present to learn a shared generative latent representation for multi-view clustering. Specifically, the generative approach assumes that the data of different views share a commonly conditional distribution of hidden variables given observed data and the hidden data are sampled independently from a mixture of Gaussian distributions.
- To better exploit the information from multiple views, we introduce a set of non-negative combination weights which will be learned jointly with the deep autoencoders network in a unified framework.
- We conduct a number of numerical experiments showing that the proposed method outperforms the state-of-the-art clustering models on several famous datasets including large-scale multi-view data.

Related Works

In literature, there are a few studies on clustering using deep neural networks (Ji et al. 2017; Peng et al. 2016; Tian et al. 2014; Xie, Girshick, and Farhadi 2016; Yang et al. 2017). In a sense, the algorithms are roughly divided into two categories, i.e., separately and jointly deep clustering approaches. The earlier deep clustering algorithms (Ji et al. 2017; Peng et al. 2016; Tian et al. 2014) often work in two stages: firstly, extracting deep features and performing traditional clustering successively, such as the K-means and spectral clustering, for the final segmentation. Yet the separated process does not help learn clustering favourable features. To this end, the jointly feature learning and clustering methods (Xie, Girshick, and Farhadi 2016; Yang et al. 2017) are proposed based on deep neural networks. In (Xie, Girshick, and Farhadi 2016), Xie *et al.* presented Deep Embedded Clustering (DEC) to learn a mapping from the data space to a lower-dimensional feature space, where it iteratively optimizes a Kullback-Leibler (KL) divergence based clustering objective. In (Yang et al. 2017), Yang *et al.* proposed a dimensionality reduction jointly with K-means clustering framework, where deep neural networks are applied to dimensionality reduction.

However, due to the limitation of the similarity measures in the aforementioned methods, the hidden, hierarchical dependencies in the latent space of data are often not able to be captured effectively. Instead, deep generative models were built to better handle the rich latent structures within data (Jiang et al. 2017). In essence, deep generative models are utilized to estimate the density of observed data under some assumptions about its latent structure, i.e., the hidden causes. Recently, Jiang *et al.* (Jiang et al. 2017) proposed a novel clustering framework, by integrating VAE and a GMM for clustering tasks, namely Variational Deep Embedding (VaDE). Unfortunately, as this method mainly focuses on single-view data, the complementary information from multiple heterogeneous views cannot be efficiently exploited. In other words, the existing generative model cannot deal with the shared latent representations for modeling the generative process of each view data.

The Proposed Method

The Architecture

Given a collection of multi-view data set $\{\mathrm{X}^{(v)} \in \mathbb{R}^{d_v \times n}\}$ ($v = 1, 2, ..., m$), totally m views, it is reasonable to assume that the i-th sample of the v-th view $\mathrm{x}_i^{(v)} \in \mathbb{R}^{d_v}$ is generated by some unknown process, for example, from an unobserved continuous variable $\mathrm{z} \in \mathbb{R}^{\mathrm{d}}$. The variable z is a common hidden representation shared by all views. Furthermore, in a typical setting, each sample $\mathrm{x}^{(v)}$ of a view is assumed to be generated through a two-stage process: first the hidden variable z is generated according to some prior distribution and then the observed sample $\mathrm{x}^{(v)}$ is yielded by some conditional distributions $p_{\theta^{(v)}}(\mathrm{x}^{(v)}|\mathrm{z})$. Usually, due to the unknown of the z and parameters θ, the prior $p_\theta(\mathrm{z})$ and the likelihood $p_{\theta^{(v)}}(\mathrm{x}^{(v)}|\mathrm{z})$ are hidden.

For clustering tasks, it is desired that the observed sample is generated jointly according to the latent variable z and an assumed clustering variable c. However, the most existing variational autoencoders are not suitable for clustering tasks by design, even to say nothing of multi-view clustering. Therefore, we are motivated to present a novel multi-view clustering under the VAE framework, by incorporating clustering-promoting objective intuitively. Ideally we shall assume that the sample generative process is given by the new likelihood $p_{\theta^{(v)}}(\mathrm{x}^{(v)}|\mathrm{z}, \mathrm{c})$, conditioned on both the hidden variable z and the cluster label c. However for simplicity we break the direct dependence of $\mathrm{x}^{(v)}$ on c conditioned on an assumed Gaussian mixture variable z. The proposed framework is shown in the right panel of Figure 1. In this architecture, multi-view samples $\{\mathrm{x}^{(v)}\}$ are generated by using DNN $f(\cdot)$ to decode the common hidden variable z, which is sampled by GMM as we assumed. To efficiently infer the posterior of both z and c from the information of multiple views, a novel weighted target distribution is introduced, based on individual variational distribution of z from each view. In order to optimize the evidence lower bound (ELBO), similar to VAE, we use DNN $g(\cdot)$ to encode observed data and incorporate the distribution of multiple embeddings to infer the shared latent representation z.

The Objective

For the sake of simplicity, we express a generic multiview variable as $\{\mathrm{x}^{(v)}\} := \{\mathrm{x}^{(1)}, ..., \mathrm{x}^{(v)}, ..., \mathrm{x}^{(m)}\}$ where $\mathrm{x}^{(v)}$ is the general variable of the v-th view. Consider the latent variables z and the discrete latent variable c ($c = 1, 2, \cdots, K$). Without loss of generality, in light of clustering task under the framework of VAE, we aim to compute the common probabilistic cluster assignment of $\{\mathrm{x}^{(v)}\}$ shared across views, denoted by $p(\mathrm{z}, c|\{\mathrm{x}^{(v)}\})$. By the Bayes theorem, the corresponding posterior of z and c given $\{\mathrm{x}^{(v)}\}$ is computed as follow.

$$p(\mathrm{z}, c|\{\mathrm{x}^{(v)}\}) = \frac{p(\{\mathrm{x}^{(v)}\}|\mathrm{z}, c)p(\mathrm{z}, c)}{\int_{\mathrm{z}} \sum_c p(\{\mathrm{x}^{(v)}\}|\mathrm{z}, c)p(\mathrm{z}, c)d\mathrm{z}}, \quad (1)$$

where we assume the views are independent, i.e., $p(\{\mathrm{x}^{(v)}\}|\mathrm{z}, c) = \prod_{v=1}^{m} p(\mathrm{x}^{(v)}|\mathrm{z}, c)$[1].

As the integral is intractable, it is hard to calculate the posterior. Inspired by the principle of VAE (Kingma and Welling 2014), we turn to compute an appropriate posterior $q(\mathrm{z}, c|\{\mathrm{x}^{(v)}\})$ to approximate the true posterior $p(\mathrm{z}, c|\{\mathrm{x}^{(v)}\})$ by minimizing the following KL divergence between them

$$\begin{aligned} & D_{KL}(q(\mathrm{z}, c|\{\mathrm{x}^{(v)}\})||p(\mathrm{z}, c|\{\mathrm{x}^{(v)}\}) \\ & = \int_{\mathrm{z}} \sum_c q(\mathrm{z}, c|\{\mathrm{x}^{(v)}\}) \log \frac{q(\mathrm{z}, c|\{\mathrm{x}^{(v)}\})}{p(\mathrm{z}, c|\{\mathrm{x}^{(v)}\})} d\mathrm{z} \\ & = -E_{q(\mathrm{z}, c|\{\mathrm{x}^{(v)}\})} \left[\log \frac{p(\{\mathrm{x}^{(v)}\}, \mathrm{z}, c)}{q(\mathrm{z}, c|\{\mathrm{x}^{(v)}\})} \right] + \log p(\{\mathrm{x}^{(v)}\}), \end{aligned} \quad (2)$$

where

$$\mathcal{L}_{\mathrm{ELBO}}(\{\mathrm{x}^{(v)}\}) \triangleq E_{q(\mathrm{z}, c|\{\mathrm{x}^{(v)}\})} \left[\log \frac{p(\{\mathrm{x}^{(v)}\}, \mathrm{z}, c)}{q(\mathrm{z}, c|\{\mathrm{x}^{(v)}\})} \right] \quad (3)$$

is called the evidence lower bound (ELBO) and $\log p(\{\mathrm{x}^{(v)}\})$ is log-likelihood.

Minimizing KL divergence is equivalent to maximizing the ELBO. Often $q(\mathrm{z}, c|\{\mathrm{x}^{(v)}\})$ is assumed to be a mean-field distribution and can be readily factorized by

$$q(\mathrm{z}, c|\{\mathrm{x}^{(v)}\}) = q(\mathrm{z}|\{\mathrm{x}^{(v)}\})q(c|\{\mathrm{x}^{(v)}\}). \quad (4)$$

Due to the powerfulness of DNN to approximate nonlinear function, we here introduce a neural network $g(\cdot)$ to infer $q(\mathrm{z}|\{\mathrm{x}^{(v)}\})$, with parameters $\{\phi^{(v)}\}_{v=1}^{m}$. That is, DNN is utilized to encode observed view data into latent representation. Meanwhile, to incorporate multi-view information, we propose a combined variational approximation $q(\mathrm{z}|\{\mathrm{x}^{(v)}\})$. Considering the importance of different views, we introduce a weight vector $\mathrm{w} = [w_1, w_2, ..., w_m]^T$ ($w_v \geq 0, \sum w_v = 1$) to fuse the distribution of hidden variables, so that the consistency and complementary of multi-view data can be better exploited. In particular, we assume the variational approximation to the posterior of latent representation z to be a Gaussian by integrating information from multiple views as follows.

$$\begin{aligned} [\tilde{\mu}^{(v)}; \log(\tilde{\sigma}^{(v)})^2] &= g(\mathrm{x}^{(v)}; \phi^{(v)}), & (5) \\ \tilde{\mu} &= \sum_{i=1}^{m} w_i \tilde{\mu}^{(i)}, & (6) \\ \tilde{\sigma}^2 &= \sum_{i=1}^{m} (w_i \tilde{\sigma}^{(i)})^2, & (7) \\ q(\mathrm{z}|\{\mathrm{x}^{(v)}\}) &= \mathcal{N}(\mathrm{z}|\tilde{\mu}, \tilde{\sigma}^2 \mathbf{I}), & (8) \end{aligned}$$

where $\mathbf{I}$ is an identity matrix with suitable dimension. In the standard VAE, each pair of $\tilde{\mu}^{(v)}$ and $(\tilde{\sigma}^{(v)})^2$ defines a Gaussian for latent variable z in the v-th view. We have fused the information in Eqs. (5) - (8).

Furthermore, ELBO can be rewritten by

$$\begin{aligned} & \mathcal{L}_{\mathrm{ELBO}}(\{\mathrm{x}^{(v)}\}) \\ & = E_{q(\mathrm{z}, c|\{\mathrm{x}^{(v)}\})} \left[\log \frac{p(\{\mathrm{x}^{(v)}\}, \mathrm{z}, c)}{q(\mathrm{z}, c|\{\mathrm{x}^{(v)}\})} \right] \\ & = \int_{\mathrm{z}} q(\mathrm{z}|\{\mathrm{x}^{(v)}\}) \log \frac{p(\{\mathrm{x}^{(v)}\}|\mathrm{z})p(\mathrm{z})}{q(\mathrm{z}|\{\mathrm{x}^{(v)}\})} d\mathrm{z} \\ & \quad - \int_{\mathrm{z}} q(\mathrm{z}|\{\mathrm{x}^{(v)}\}) D_{KL}(q(c|\{\mathrm{x}^{(v)}\})||p(c|\mathrm{z}))d\mathrm{z}, \end{aligned} \quad (9)$$

Hence, we set $D_{KL}(q(c|\{\mathrm{x}^{(v)}\})||p(c|\mathrm{z})) \equiv 0$ to maximize $\mathcal{L}_{\mathrm{ELBO}}(\{\mathrm{x}^{(v)}\})$, due to the first term has no relationship with c and the second term is non-negative. As a result, we use the following equation to compute $q(c|\{\mathrm{x}^{(v)}\})$, i.e.,

$$q(c|\{\mathrm{x}^{(v)}\}) = p(c|\mathrm{z}) \equiv \frac{p(c)p(\mathrm{z}|c)}{\sum_c p(c)p(\mathrm{z}|c)}. \quad (10)$$

This means we are proposing a mixture model for the latent prior $p(\mathrm{z})$. Particularly we implement the latent prior $p(\mathrm{z})$ as a Gaussian mixture as follows,

$$p(c) = Cat(c|\pi), \quad p(\mathrm{z}|c) = \mathcal{N}(\mathrm{z}|\mu_c, \sigma_c^2 \mathbf{I}),$$

where $Cat(c|\pi)$ is the categorical distribution with parameter $\pi = (\pi_1, ..., \pi_K) \in \mathbb{R}_+^K$, $\sum \pi_c = 1$ such that π_c ($c = 1, ..., K$) is the prior probability for cluster c, and both μ_c and σ_c^2 ($c = 1, ..., K$) are the mean and the variance of the c-th Gaussian component, respectively.

Once the latent variable z is produced according to the GMM prior, the multi-view data generative process will defined by, for the binary observed data,

$$\mu_{\theta^{(v)}} = f(\mathrm{z}; \theta^{(v)}), \quad p(\mathrm{x}^{(v)}|\mathrm{z}) = \mathrm{Ber}(\mathrm{x}^{(v)}|\mu_{\theta^{(v)}}),$$

[1] Hereafter the model parameter $\theta^{(v)}$ is omitted.

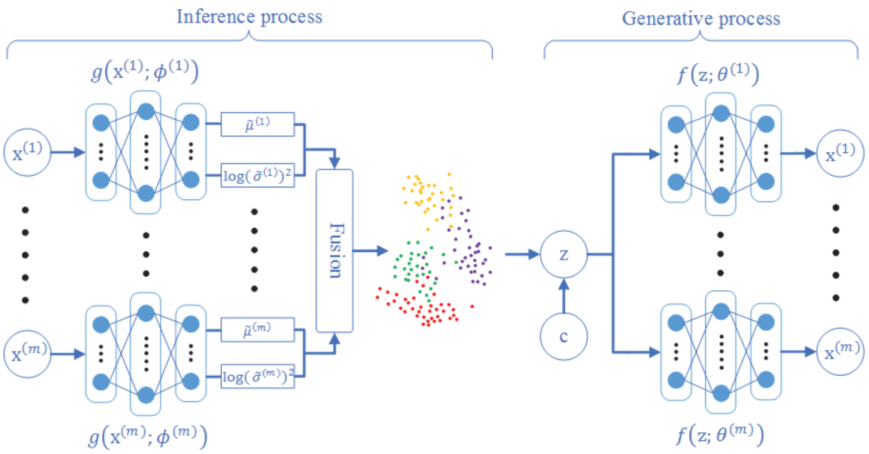


Figure 1: The architecture of the proposed multi-view model (Best seen on screen.). The data generative process under the deep autoencoders framework is performed in three steps. (a). A cluster c is first picked from the categorical distribution; (b). A shared latent representation z is sampled by GMM model coresponding to the prior picked cluster; (c). DNN $f(z; \theta^{(v)})$ decodes the latent embedding into an observable $x^{(v)}$. In the inference process, the encoder network $g(\cdot)$ and weighted fusion to variational distribution are applied to infer the posterior of both z and c from the information of multiple views.

where $f(z; \theta^{(v)})$ is a deep neural network whose input is z parameterized by $\theta^{(v)}$, and $\text{Ber}(\mu_{\theta^{(v)}})$ is multivariate Bernoulli distribution parameterized by $\mu_{\theta^{(v)}}$. Or for the continuous data,

$$\mu_{\theta^{(v)}} = f_1(z; \theta^{(v)}), \tag{11}$$

$$\log(\sigma^2_{\theta^{(v)}}) = f_2(z; \theta^{(v)}), \tag{12}$$

$$p(x^{(v)}|z) = \mathcal{N}(x^{(v)}|\mu_{\theta^{(v)}}, \sigma^2_{\theta^{(v)}}), \tag{13}$$

where $f_1(\cdot)$ and $f_2(\cdot)$ are all deep neural networks with appropriate parameters $\theta^{(v)}$, producing the mean and variance for the Gaussian likelihoods. The generative process is depicted in the right part of Figure 1.

For the v-th view, since $x^{(v)}$ and c are independent conditioned on z, the joint probability $p(\{x^{(v)}\}, z, c)$ can be decomposed by,

$$p(\{x^{(v)}\}, z, c) == p(\{x^{(v)}\}|z)p(z|c)p(c). \tag{14}$$

Next, to use the Stochastic Gradient Variational Bayes (SGVB) (Kingma and Welling 2014), by using the *reparameterization* trick,the objective function of our method can be formulated by,

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\{x^{(v)}\}) &= \sum_{c=1}^{K} \gamma_c \log \frac{\pi_c}{\gamma_c} + \frac{1}{2}\sum_{j=1}^{J}(1 + \log \tilde{\sigma}^2|_j) \\ &+ \frac{1}{L}\sum_{v=1}^{m}\sum_{\iota=1}^{L}\sum_{i=1}^{D} x_i^{(v)} \log \mu_{\theta^{(v)}}|_i^l + (1 - x_i^{(v)}) \log(1 - \mu_{\theta^{(v)}}|_i^l) \\ &- \frac{1}{2}\sum_{c=1}^{K}\gamma_c \sum_{j=1}^{J}(\log \sigma_c^2|_j + \frac{\tilde{\sigma}^2|_j}{\sigma_c^2|_j} + \frac{(\tilde{\mu}|_j - \mu_c|_j)^2}{\sigma_c^2|_j}),\end{aligned} \tag{15}$$

where $\mu_{\theta^{(v)}}$ is outputs of the DNN $f(\cdot)$, L denotes the number of Monte Carlo samples in the SGVB estimator and is usually set to be 1. The dimension for $x^{(v)}$ and $\mu_{\theta^{(v)}}$ is D while the dimension for $\mu_c, \tilde{\mu}, \sigma_c^2$ and $\tilde{\sigma}^2$ is J. $x_i^{(v)}$ denotes the i-th element of $x^{(v)}$, $*|_i^l$ represents l-th sample and i-th element of $*$, and $*|_j$ means the j-th element of $*$. γ_c denotes $q(c|\{x^{(v)}\})$ for simplicity.

For the continuous data, the objective function is rewritten as:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\{x^{(v)}\}) &= \sum_{c=1}^{K} \gamma_c \log \frac{\pi_c}{\gamma_c} + \frac{1}{2}\sum_{j=1}^{J}(1 + \log \tilde{\sigma}^2|_j) \\ &+ \frac{1}{L}\sum_{v=1}^{m}\sum_{\iota=1}^{L}\sum_{i=1}^{D} -\frac{1}{2}\log 2\pi\sigma^2_{\theta^{(v)}}|_i^l - \frac{(x_i^{(v)} - \mu_{\theta^{(v)}}|_i^l)^2}{2\sigma^2_{\theta^{(v)}}|_i^l} \\ &- \frac{1}{2}\sum_{c=1}^{K}\gamma_c \sum_{j=1}^{J}(\log \sigma_c^2|_j + \frac{\tilde{\sigma}^2|_j}{\sigma_c^2|_j} + \frac{(\tilde{\mu}|_j - \mu_c|_j)^2}{\sigma_c^2|_j}),\end{aligned} \tag{16}$$

where $\mu_{\theta^{(v)}}$ and $\sigma^2_{\theta^{(v)}}$ can be obtained by Eq. (11) and Eq. (12), respectively. Intuitively, the third term of Eq. (16) is used for reconstruction, and the rest is the KL divergence from the Gaussian mixture prior $p(z, c)$ to the variational posterior $q(z, c|\{x^{(v)}\})$. As such, the model can not only generate the samples well, but make variational inference close to our hypothesis.

Remark: Note that although our model is also equipped with VAE and GMM, it is distinct from the existing work (Du, Du, and He 2017; Jiang et al. 2017). Our model focuses on multi-view clustering task by simultaneously learning the generative network, inference network and the weight of each view.

By a direct application of the chain rule and estimators, similar to the work (Du, Du, and He 2017; Jiang et al. 2017), the gradients of the loss for Eq. (15) is calculated readily. To train the model, the estimated gradients in conjunction

Table 1: Dataset Summary

Datasets	# of samples	# of views	# of classes
UCI digits	2,000	6	10
Caltech-7	1,474	6	7
ORL	400	3	40
NUS-WIDE-Object	30,000	5	31

with standard stochastic gradient based optimization methods, such as SGD or Adam, are applied. Overall, the proposed model can be trained with *reparameterization* trick for back-propagation through the mixed Gaussian latent variables. After training, the shared latent representation z is achieved for each sample $\mathrm{x}_i (i = 1, 2, ..., n)$. Finally the final cluster assignment is computed by Eq. (10).

Experimental Results

Datasets

To evaluate the performance of the proposed DMVCVAE, we select four real-world datasets including digits, object and facial images. A summary of the dataset statistics is also provided in Table 1.

- **UCI digits**[2] consists of features of handwritten digits of 0 to 9 extracted from UCI machine learning repository. It contains 2000 data points with 200 samples for each digit. These digits are represented by six types of features, including pixel averages in 2×3 windows (PIX) of dimension 240, Fourier coefficients of the character shapes-dimension 76, profile correlations (FAC) of dimension 216, Zernike moments (ZER) of dimension 47, Karhunen-Loeve coefficients (KAR) of dimension 64 and morphological features (MOR) of dimension 6.
- **Caltech 101** is an object recognition dataset (Li, Fergus, and Perona 2004) containing 8677 images of 101 categories. We chose 7 classes of Caltech 101 with 1474 images, i.e., Face, Motorbikes, Dolla-Bill, Garfield, Snoopy, Stop-Sign and Windsor-Chair. There are six different views, including Gabor features of dimension of 48, wavelet moments of dimension 40, CENTRIST features of dimension 254, histogram of oriented gradients(HOG) of dimension 1984, GIST features of dimension 512, and local binary patterns (LBP) of dimension 928.
- **ORL** contains 10 different images from each of 40 distinct subjects. For some subjects, the images were taken at different times with varying lighting, facial expressions and facial details. It consists of three types of features: intensity of dimension 4096, LBP features of dimension 3304 and Gabor features of dimension 6750.
- **NUS-WIDE-Object (NUS)** is a dataset for object recognition which consists of 30000 images in 31 classes. We use 5 features provided by the web-site, i.e. 65 dimension color Histogram (CH), 226 dimension color moments (CM), 145 dimension color correlation (CORR), 74 dimension edge distribution and 129 wavelet texture.

Experiment Settings

In our experiments, the fully connected network and same architecture settings as DEC (Xie, Girshick, and Farhadi 2016) are used. More specifically, the architectures of $g(\mathrm{x}^{(v)}; \phi^{(v)})$ and $f(\mathrm{z}; \theta^{(v)})$ are d_v-500-500-200-10 and 10-2000-500-500-d_v, respectively, where d_v is input dimensionality of each view. Here, other architectures such as Convolutional Neural Network (CNN) and Deep Belief Network (DBN) are also viable options. We use Adam optimizer (Kingma and Ba 2015) to maximize the objective function, and set the learning rate to be 0.0001 with a decay of 0.9 for every 10 epochs.

Initializing the parameters of the deep neural network is usually utilized to avoid the problem that the model might get stuck in a undesirable local minima or saddle points. Here, we use layer-wise pre-training method (Bengio et al. 2007) for training DNN $g(\cdot)$ and $f(\cdot)$. After pre-training, the network $g(\cdot)$ is adopted to project input data points into the latent representation z, and then we perform K-means to z to obtain K initial centroids of GMM $\mu_c (c \in \{1, \cdots, K\})$. Besides, the weights w of Eqs. (6) and (7) are initialized to $\frac{1}{m}$ for each view and the parameter of GMM π_k is initialized to $\frac{1}{K}$.

Three popular metrics are used to evaluate the clustering performance, i.e. clustering accuracy (ACC), normalized mutual information (NMI) and adjusted rand index (ARI), in which the clustering accuracy is defined by

$$\mathrm{ACC} = \max_{m \in \mathcal{M}} \frac{\sum_{i=1}^{N} 1\{l_i = m(c_i)\}}{N},$$

where l_i is the ground-truth label, c_i is the cluster assignment obtained by the model, and $\mathcal{M}$ ranges over all possible one-to-one mappings between cluster assignment and labels. The mapping $m(\cdot)$ can be efficiently fulfilled by the Kuhn-Munkres algorithm (Chen, Donoho, and Saunders 2001). NMI indicates the correlation between predicted labels and ground truth labels. ARI scales from -1 to 1, which measures the similarity between two data clusterings, higher value usually means better clustering performance. As each measure penalizes or favors different properties in the clustering, we report results on all the measures for a comprehensive evaluation.

Baseline Algorithms

We compare the proposed DMVCVAE with the following clustering methods including both shallow models and deep models.

- *Single View*: Choosing the single view of the best clustering performance using the graph Laplacian derived from and performing spectral clustering on it.
- *Feature Concatenation* (abbreviated to Feature Concat.): Concatenating the features of all views and conducting spectral clustering on it.
- *Kernel Addition*: Building an affinity matrix from every feature and taking an average of them, then inputting to a spectral clustering algorithm.

[2] https://archive.ics.uci.edu/ml/datasets/Multiple+Features

- *MultiNMF*(Liu et al. 2013): *Multi-view NMF* 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.
- *LT-MSC*(Zhang et al. 2015): *Low-rank tensor constrained multi-view subspace clustering* proposes a multi-view clustering by considering the subspace representation matrices of different views as a tensor.
- *SCMV-3DT*(Yin et al. 2019): *Low-rank multi-view clustering in third-order tensor space via t-linear combination* using *t-product* based on the circular convolution to reconstruct multi-view tensorial data by itself with sparse and low-rank penalty.
- *DCCA* (Andrew et al. 2013): Providing flexible nonlinear representations with respect to the correlation objective measured on unseen data.
- *DCCAE* (Wang et al. 2015): Combining the DCCA objective and reconstruction errors of the two views.
- *VCCAP* (Wang et al. 2016): Using a deep generative method to achieve a natural idea that the multiple views can be generated from a small set of shared latent variables.

Performance Evaluation

We first compare our method with six shallow models on the chosen test datasets. The parameter settings for the compared methods are done according to their authors' suggestions for their best clustering scores. The clustering performance of different methods are achieved by running 10 trials and reporting the average score of the performance measures, shown in Table 2. The bold numbers highlight the best results.

As can be seen, except for the *Single View*, the other methods exploit all of views data with an improved performance than using a single view.In terms of all of these evaluation criteria, our proposed method consistently outperforms the shallow models for UCI digits and Caltech-7 datasets. In particularly, for Caltech-7, our method outperforms the second best algorithm in terms of ACC and NMI by 17.7% and 25.0%, respectively. While for ORL dataset, LT-MSC and SCMV-3DT achieves the best result in terms of NMI and ARI, respectively. This may be explained by the small size of ORL dataset, since large-scale datasets often lead to better performance for deep models. The results also verify that our model DMVCVAE significantly benefits from deep learning.

To further verify the performance of our approach among the deep models, we report the comparisons between the deep models, given in Table 3. Since these three models can only handle two views data, we tested all the two view combination and the best clustering score is reported finally. The hyper-parameters of the compared models are suggested by their papers. Specifically, FAC and KAR features are chosen in UCI digits, GIST and LBP features for Caltech-7, and LBP and Gabor features for ORL. For fair comparison, we perform the proposed model on the same views. From Table 3, it is observed that our proposed method significantly outperforms others on all criteria.

Table 2: Clustering performance comparison between the propose model and shallows methods.

Methods	UCI-digits			Caltech-7			ORL		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
BestView	0.6956	0.6424	0.7301	0.4100	0.4119	0.2582	0.6700	0.8477	0.5676
Feature Concat.	0.6973	0.6973	0.6064	0.3800	0.3410	0.2048	0.6700	0.8329	0.5590
Kernel Addition	0.7700	0.7456	0.3700	0.3936	0.2573	0.6570	0.6000	0.8062	0.4797
MultiNMF	0.7760	0.7041	0.6031	0.3602	0.3156	0.1965	0.6825	0.8393	0.5736
LT-MSC	0.8422	0.8217	0.7584	0.5665	0.5914	0.4182	0.7587	**0.9094**	0.7093
SCMV-3DT	0.9300	0.8608	0.8459	0.6246	0.6031	0.4693	0.7947	0.9088	**0.7381**
Ours	**0.9570**	**0.9166**	**0.9107**	**0.8014**	**0.8538**	**0.7048**	**0.7975**	0.9013	0.7254

Table 3: Clustering performance comparison among the deep models.

Methods	UCI-digits			Caltech-7			ORL		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
DCCA	0.8195	0.8020	0.7424	0.8242	0.6781	0.7131	0.6125	0.8094	0.4699
DCCAE	0.8205	0.8057	0.7458	0.8462	0.7054	0.7319	0.6425	0,8115	0.5048
VCCAP	0.7480	0.7320	0.6277	0.8372	0.6301	0.7206	0.4150	0.6440	0.2418
Ours	**0.8875**	**0.8076**	**0.7765**	**0.8568**	**0.7386**	**0.7826**	**0.6950**	**0.8356**	**0.5643**

Relation of our work to VaDE

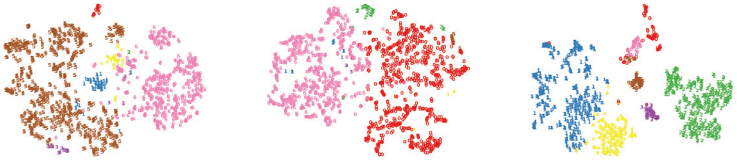
Our work is inspired by the Variantional Deep Embedding (VaDE) (Jiang et al. 2017), which mainly focuses the clustering for single-view data. However, VaDE cannot be *directly* utilized for multi-view data given its natural structure. Thus in this subsection, we will comprehensively compare ours with VaDE on the datasets, by applying VaDE to single view and concatenated feature respectively. In particular, applying VaDE to single view is to use each view as input, while the concatenated feature is attained by stacking all views to be one long vector, such that the task is performed by single-view clustering. The results are reported in Tables 4- 6. As can be seen, our method achieves the best performance in terms of all measures. For UCI-digits dataset, the score of VaDE with PIX in Single View is the second best. Note that PIX view represents the pixel values of raw images. The similar cases are HOG for Caltech-7 and Intensity feature for ORL respectively. Meanwhile, the performance of Feature Concatenation is even worse than that of using single view. This demonstrates that it is not a feasible way to directly apply VaDE to multi-view clustering. A superior approach is verified to be fully aware of consistency and complementary information from all views.

Visualizations

In Figure 2, we visualize the latent space on Caltech-7 dataset by various deep models. t-SNE (Maaten and Hinton 2008) is applied to reducing the dimensionality to 2-dimensional space. It can be observed that the embedding learned by DMVCVAE is better than that by DCCAE and VCCAP. Figure 3 shows the learned representations of DMVCVAE on UCI digits dataset. Specifically, we see that,

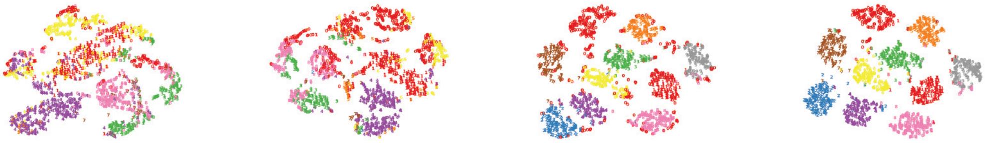
Table 4: Clustering performance for VaDE on UCI-digits.

UCI-digits	PIX	FOU	FAC	ZER	KAR	MOR	Feat.	Ours
ACC	0.8341	0.3741	0.7961	0.3385	0.4321	0.2891	0.8055	**0.9570**
NMI	0.7211	0.2233	0.6771	0.1748	0.2547	0.4958	0.7454	**0.9166**
ARI	0.6765	0.1498	0.6081	0.1062	0.1735	0.2545	0.6682	**0.9107**



(a) DCCAE (b) VCCAP (c) DMVCVAE

Figure 2: Visualization to show the latent subspaces of Caltech-7 dataset.



(a) Epoch 10 (b) Epoch 40 (c) Epoch 70 (d) Epoch 100

Figure 3: Visualization to show the latent subspaces of UCI digits by DMVCVAE visualization from epoch 10 to 100.

Table 5: Clustering performance for VaDE on Caltech-7.

Caltech-7	Gabor	wavelet	CENT.	HOG	GIST	LBP	Feat.	Ours
ACC	0.3868	0.4981	0.4825	0.6078	0.4954	0.4689	0.5752	**0.8014**
NMI	0.4262	0.5440	0.5270	0.6427	0.5447	0.5220	0.6143	**0.8538**
ARI	0.3130	0.3934	0.3666	0.4979	0.4026	0.3884	0.4620	**0.7048**

as training progressing, the latent feature clusters become more and more separated, suggesting that the overall architecture motivates seeking informative representations with better clustering performance.

Experiment on large-scale multi-view data

With the unprecedentedly explosive growth in the volume of visual data, how to effectively segment large-scale multi-view data becomes an interesting but challenging problem (Li et al. 2015; Zhang et al. 2019). Therefore, we further test our model on the large-scale dataset, i.e., NUS-WIDE-Object. As the aforementioned compared methods cannot handle the large-scale data, we compare with the recent work, such as Large-Scale Multi-View Spectral Clustering (LSMVSC) (Li et al. 2015) and Binary Multi-View Clustering (BMVC) (Zhang et al. 2019). In this experiment, we replace the ARI measure with PURITY such that the comparison will be fair[3]. By the similar settings, the clustering results are reported in Table 7. As can be seen, our proposed approach achieved better clustering performance against the compared ones and verified the strong capacity on handling large-scale multi-view clustering.

Table 6: Clustering performance for VaDE on ORL.

ORL	Intensity	LBP	Gabor	Feat.	Ours
ACC	0.5250	0.4175	0.3260	0.4845	**0.7975**
NMI	0.7081	0.5612	0.5058	0.6958	**0.9013**
ARI	0.4127	0.3650	0.2486	0.3725	**0.7254**

Table 7: Clustering performance for large-scale dataset.

Methods	NUS-WIDE-Object		
	ACC	NMI	PURITY
LSMVSC	–	0.1493	0.2821
BMVC	0.1680	0.1621	0.2872
Ours	**0.1909**	**0.2129**	**0.3168**

Conclusions

In this paper, we proposed a novel multi-view clustering algorithm by learning a shared latent representation under the VAE framework. The shared latent embeddings, multi-view weights and deep autoencoders networks are simultaneously learned in a unified framework such that the final clustering assignment is intuitively achieved. Experimental results show that the proposed method can provide better clustering solutions than other state-of-the-art approaches, including the shallow models and deep models.

References

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

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

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *ECCV*.

Chang, J.; Wang, L.; Meng, G.; Xiang, S.; and Pan, C. 2017. Deep adaptive image clustering. In *ICCV*.

Chaudhuri, K.; Kakade, S. M.; Livescu, K.; and Sridharan,

[3] Here we presented the best reported results from their original papers.

K. 2009. Multi-view clustering via canonical correlation analysis. In *ICML*.

Chen, S. S.; Donoho, D. L.; and Saunders, M. A. 2001. Atomic decomposition by basis pursuit. *SIAM Review* 43(1):129–159.

Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *CVPR*.

Deng, J.; Dong, W.; Socher, R.; jia Li, L.; Li, K.; and Feifei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Du, C.; Du, C.; and He, H. 2017. Sharing deep generative representation for perceived image reconstruction from human brain activity. In *IJCNN*.

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

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

Ji, P.; Zhang, T.; Li, H.; Salzmann, M.; and Reid, I. 2017. Deep subspace clustering networks. In *NIPS*, 24–33.

Jiang, Z.; Zheng, Y.; Tan, H.; Tang, B.; and Zhou, H. 2017. Variational deep embedding: An unsupervised and generative approach to clustering. In *IJCAI*.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. *CoRR* abs/1312.6114.

Li, Y.; Nie, F.; Huang, H.; and Huang, J. 2015. Large-scale multi-view spectral clustering via bipartite graph. In *AAAI*.

Li, F.-F.; Fergus, R.; and Perona, P. 2004. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *CVPR Workshop*.

Liu, J.; Wang, C.; Gao, J.; and Han, J. 2013. Multi-view clustering via joint nonnegative matrix factorization. In *SIAM Data Mining*.

Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-SNE. *JMLR* 9:2579–2605.

Ngiam, J.; Khosla, A.; Kim, M.; Nam, J.; Lee, H.; and Ng, A. Y. 2011. Multimodal deep learning. In *ICML*.

Ojala, T.; Pietikainen, M.; and Maenpaa, T. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE PAMI* 24(7):971–987.

Peng, X.; Xiao, S.; Feng, J.; Yau, W.-Y.; and Yi, Z. 2016. Deep subspace clustering with sparsity prior. In *IJCAI*.

Pu, Y.; Gan, Z.; Henao, R.; Yuan, X.; Li, C.; Stevens, A.; and Carin, L. 2016. Variational autoencoder for deep learning of images, labels and captions. In *NIPS*.

Srivastava, N., and Salakhutdinov, R. 2014. Multimodal learning with deep boltzmann machines. *JMLR* 15(1):2949–2980.

Sun, S. 2013. A survey of multi-view machine learning. *Neural Comput and Appl* 23(7):2031–2038.

Tian, F.; Gao, B.; Cui, Q.; Chen, E.; and Liu, T.-Y. 2014. Learning deep representations for graph clustering. In *AAAI*.

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

Wang, W.; Yan, X.; Lee, H.; and Livescu, K. 2016. Deep variational canonical correlation analysis. *arXiv:1610.03454*.

Wang, H.; Nie, F.; and Huang, H. 2013. Multi-view clustering and feature learning via structured sparsity. In *ICML*.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *ICML*, 478–487.

Xu, J.; Han, J.; Nie, F.; and Li, X. 2017. Re-weighted discriminatively embedded k-means for multi-view clustering. *IEEE TIP* 26(6):3016–3027.

Xu, C.; Guan, Z.; Zhao, W.; Niu, Y.; Wang, Q.; and Wang, Z. 2018. Deep multi-view concept learning. In *IJCAI*.

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

Yang, B.; Fu, X.; Sidiropoulos, N. D.; and Hong, M. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *ICML*.

Yin, M.; Gao, J.; Xie, S.; and Guo, Y. 2019. Multiview subspace clustering via tensorial t-product representation. *IEEE NNLS* 30(3):851–864.

Zhang, C.; Fu, H.; Liu, S.; Liu, G.; and Cao, X. 2015. Low-rank tensor constrained multiview subspace clustering. In *ICCV*.

Zhang, Z.; Liu, L.; Shen, F.; Shen, H. T.; and Shao, L. 2019. Binary multi-view clustering. *IEEE PMAI* 41(7):1774–1782.

Bengio, Y.; Lamblin, P.; Popovici, D.; and Larochelle, H. 2007. Greedy layer-wise training of deep networks. In *NIPS* 153–160.