Variational Deep Embedding: A Generative Approach to Clustering

Zhuxi Jiang
Beijing Institute of Technology
Beijing 100081, China
zjiang@bit.edu.cn

Yin Zheng
Hulu LLC.
Beijing, China, 100084
yin.zheng@hulu.com

Huachun Tan
Beijing Institute of Technology
Beijing 100081, China
tanhc@bit.edu.cn

Bangsheng Tang
Hulu LLC.
Beijing, China, 100084
bangsheng@hulu.com

Hanning Zhou
Hulu LLC.
Beijing, China, 100084
eric.zhou@hulu.com

Abstract

Clustering is among the most fundamental tasks in computer vision and machine learning. In this paper, we propose Variational Deep Embedding (VaDE), a novel unsupervised generative clustering approach within the framework of Variational Auto-Encoder (VAE). Specifically, VaDE models the data generative procedure with a Gaussian Mixture Model (GMM) and a deep neural network (DNN): 1) the GMM picks a cluster; 2) from which a latent embedding is generated; 3) then the DNN decodes the latent embedding into an observable. Inference in VaDE is done in a variational way: a different DNN is used to encode observables to latent embeddings, so that the evidence lower bound (ELBO) can be optimized using Stochastic Gradient Variational Bayes (SGVB) estimator and the reparameterization trick. Quantitative comparisons with strong baselines are included in this paper, and experimental results show that VaDE significantly outperforms the state-of-the-art clustering methods on 4 *benchmarks from various modalities. Moreover, by VaDE's generative nature, we show its capability of generating highly realistic samples for any specified cluster, without using supervised information during training. Lastly, VaDE is a flexible and extensible framework for unsupervised generative clustering, more general mixture models than GMM can be easily plugged in.*

1. Introduction

Clustering is the process of grouping similar objects together, which is one of the most fundamental tasks in computer vision and machine learning. Over the past decades, a large family of clustering algorithms have been developed and successfully applied in enormous real world tasks [26, 20, 34, 33, 32, 21, 31]. Generally speaking, there is a dichotomy of clustering methods: Similarity-based clustering and Feature-based clustering. Similarity-based clustering builds models upon a distance matrix, which is a $N \times N$ matrix that measures the distance between each pair of the N samples. One of the most famous similarity-based clustering methods is Spectral Clustering (SC) [30, 32, 21], which leverages the Laplacian spectra of the distance matrix to reduce dimensionality before clustering. Similarity-based clustering methods have the advantage that domain-specific similarity or kernel functions can be easily incorporated into the models. But these methods suffer scalability issue due to super-quadratic running time for computing spectra.

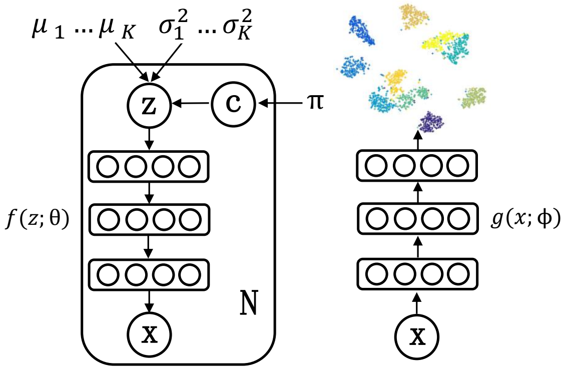


Figure 1. The diagram of VaDE. The data generative process of VaDE is done as follows: 1) a cluster is picked from a GMM model; 2) a latent embedding is generated based on the picked cluster; 3) DNN $f(\mathbf{z};\boldsymbol{\theta})$ decodes the latent embedding into an observable $\mathbf{x}$. A encoder network $g(\mathbf{x};\phi)$ is used to maximize the ELBO of VaDE.

Different from similarity-based methods, a feature-based method takes a $N \times D$ matrix as input, where N is the number of samples and D is the feature dimension. One popular feature-based clustering method is K-means, which aims to partition the samples into K clusters so as to minimize the within-cluster sum of squared errors. Another representative feature-based clustering model is Gaussian Mixture Model (GMM), which assumes that the data points are generated from a Mixture-of-Gaussians (MoG), and the parameters of GMM are optimized by the Expectation Maximization (EM) algorithm. One advantage of GMM over K-means is that a GMM can generate samples by estimation of data density. Although K-means, GMM and their variants [33, 27, 25, 38, 15, 8] have been extensively used, learning good representations most suitable for clustering tasks is left largely unexplored.

Recently, deep learning has achieved widespread success in numerous machine learning tasks [9, 11, 3, 36, 35, 37, 1, 7], where learning good representations by deep neural networks (DNN) lies in the core. Taking a similar approach, it is conceivable to conduct clustering analysis on good representations, instead of raw data points. In a recent work, Deep Embedded Clustering (DEC) [31] was proposed to simultaneously learn feature representations and cluster assignments by deep neural networks. Although DEC performs well in clustering, similar to K-means, DEC cannot generate samples. Some recent works, e.g. VAE [10], GAN [5] and PixelRNN [22], have shown that neural networks can be trained to generate meaningful samples. The motivation of this work is to develop a unified model that 1) captures the statistical structure of the data, and 2) is capable of generating samples.

In this paper, we propose a framework, Variational Deep Embedding (VaDE), that combines VAE [10] and probabilistic mixture models. VaDE models the data generative process by a GMM and a DNN f: 1) a cluster is picked up by GMM; 2) from which a latent representation $\mathbf{z}$ is sampled; 3) DNN f decodes $\mathbf{z}$ to an observation $\mathbf{x}$. Moreover, VaDE is optimized by using another DNN g to encode observed data $\mathbf{x}$ into latent embedding $\mathbf{z}$, so that the Stochastic Gradient Variational Bayes (SGVB) estimator and the *reparameterization* trick [10] can be used to maximize the evidence lower bound (ELBO). VaDE generalizes VAE [10] in that a Mixture-of-Gaussians prior replaces the single Gaussian prior in VAE. Hence, VaDE is by design more suitable for clustering tasks. Note that VaDE is a flexible unsupervised generative clustering framework, and other probabilistic mixture models [19] can be easily plugged in. The diagram of VaDE is illustrated in Figure 1.

The main contributions of the paper are:

- We propose a generic framework, VaDE, for clustering tasks, generalizing VAE and probabilistic mixture models;
- We show how to optimize VaDE by maximizing the ELBO using the SGVB estimator and the *reparameterization* trick;
- Experimental results show that VaDE outperforms the state-of-the-art clustering models on 4 datasets from various modalities by a large margin;
- We show that VaDE can generate highly realistic samples for any specified cluster, without using supervised information during training.

2. Related Work

The goal of deep generative models is to estimate density of data by neural networks, from which unseen samples can be generated. Recently, the most famous and successful deep generative models are GAN [5] and VAE [10]. GAN models the generative process as an adversarial process, where a generative model G tries to generate samples from the data distribution, while a discriminative model D learns to determine whether a sample is from the model distribution or data distribution. Different from the adversarial process of GAN, VAE models the generative process by a single Gaussian prior in latent space and a decoder network. An encoder network is utilized to maximized the ELBO of VAE by the SGVB estimator and the *reparameterization* trick. Both GAN and VAE are appealing unsupervised generative models because they can be optimized by stochastic gradient decent and generate samples from complicated distributions [6, 17, 4, 24, 12].

Recently, DEC [31] was proposed to learn feature representations and cluster assignments simultaneously using deep neural networks. In fact, DEC learns a mapping from the observed space to a lower-dimensional latent space, where it iteratively optimizes the KL divergence. DEC achieved impressive performances on clustering tasks. However, the assumptions underlying the feature embedding and the clustering are generally independent, which is the main unsatisfactory point of the model. In this work, we propose VaDE, an unsupervised clustering method that generalizes VAE and probabilistic mixture models. Different from DEC, the latent lower-dimensional representations learned by VaDE are encouraged to approximate the Mixture-of-Gaussians prior, which are suitable for clustering tasks by design.

3. Variational Deep Embedding

In this section, we describe Variational Deep Embedding (VaDE), a model for probabilistic clustering problem within the framework of Variational Auto-Encoder (VAE). Throughout this paper, we assume Mixture-of-Gaussians as the prior of the probabilistic clustering, for its generality and simplicity. This assumption is not fundamental, since other probabilistic mixture models can be easily plugged in this framework.

3.1. The Generative Process

Here we describe the generative process of VaDE. Specifically, suppose there are K clusters, an observed sample $\mathbf{x} \in \mathbb{R}^D$ is generated by the following process:

1. Choose a cluster $c \sim \text{Cat}(\boldsymbol{\pi})$
2. Choose a latent vector $\mathbf{z} \sim \mathcal{N}\left(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2\mathbf{I}\right)$
3. Choose a sample $\mathbf{x}$:
   (a) If $\mathbf{x}$ is binary
      i. Compute the expectation vector $\boldsymbol{\mu}_x$
      $$\boldsymbol{\mu}_x = f(\mathbf{z};\boldsymbol{\theta}) \tag{1}$$
      ii. Choose a sample $\mathbf{x} \sim \text{Ber}(\boldsymbol{\mu}_x)$
   (b) If $\mathbf{x}$ is real-valued
      i. Compute $\boldsymbol{\mu}_x$ and $\boldsymbol{\sigma}_x^2$
      $$[\boldsymbol{\mu}_x; \log\boldsymbol{\sigma}_x^2] = f(\mathbf{z};\boldsymbol{\theta}) \tag{2}$$
      ii. Choose a sample $\mathbf{x} \sim \mathcal{N}\left(\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x^2\mathbf{I}\right)$

where K is a predefined parameter, π_k is the prior probability for cluster k, $\boldsymbol{\pi} \in \mathbb{R}_+^K$, $1 = \sum_{k=1}^K \pi_k$, $\text{Cat}(\boldsymbol{\pi})$ is the categorical distribution parametrized by $\boldsymbol{\pi}$, $\boldsymbol{\mu}_c$ and $\boldsymbol{\sigma}_c^2$ are the mean and the variance of the Gaussian distribution corresponding to cluster c, $\mathbf{I}$ is an identity matrix, $f(\mathbf{z};\boldsymbol{\theta})$ is a neural network whose input is $\mathbf{z}$ and is parametrized by $\boldsymbol{\theta}$, $\text{Ber}(\boldsymbol{\mu}_x)$ and $\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x^2)$ are multivariate Bernoulli distribution and Gaussian distribution parametrized by $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x$, respectively. The generative process is depicted in Figure 1.

According to the generative process above, the joint probability $p(\mathbf{x}, \mathbf{z}, c)$ can be factorized as:

$$p(\mathbf{x}, \mathbf{z}, c) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}|c)p(c), \tag{3}$$

since $\mathbf{x}$ and c are independent conditioned on $\mathbf{z}$. And the probabilities are defined as:

$$\begin{aligned} p(c) &= \text{Cat}(c|\boldsymbol{\pi}) && (4)\\ p(\mathbf{z}|c) &= \mathcal{N}\left(\mathbf{z}|\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2\mathbf{I}\right) && (5)\\ p(\mathbf{x}|\mathbf{z}) &= \text{Ber}(\mathbf{x}|\mu_x) \quad \textit{or} \quad \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x^2\mathbf{I}) && (6) \end{aligned}$$

3.2. Variational Lower Bound

A VaDE instance is tuned to maximize the likelihood of the given data points. Given the generative process in Section 3.1, the log-likelihood of VaDE can be written as:

$$\begin{aligned} \log p(\mathbf{x}) &= \log \int_{\mathbf{z}} \sum_c p(\mathbf{x}, \mathbf{z}, c) d\mathbf{z} \\ &= \log \int_{\mathbf{z}} \sum_c q(\mathbf{z}, c|\mathbf{x}) \frac{p(\mathbf{x}, \mathbf{z}, c)}{q(\mathbf{z}, c|\mathbf{x})} d\mathbf{z} \\ &\geq \int_{\mathbf{z}} \sum_c q(\mathbf{z}, c|\mathbf{x}) \log \frac{p(\mathbf{x}, \mathbf{z}, c)}{q(\mathbf{z}, c|\mathbf{x})} d\mathbf{z} \\ &= E_{q(\mathbf{z},c|\mathbf{x})}[\log \frac{p(\mathbf{x}, \mathbf{z}, c)}{q(\mathbf{z}, c|\mathbf{x})}] = \mathcal{L}_{\text{ELBO}}(\mathbf{x}) \end{aligned} \tag{7}$$

where $\mathcal{L}_{\text{ELBO}}$ is the evidence lower bound (ELBO), $q(\mathbf{z}, c|\mathbf{x})$ is the variational posterior to approximate the true posterior $p(\mathbf{z}, c|\mathbf{x})$, and we assume it can be factorized as:

$$q(\mathbf{z}, c|\mathbf{x}) = q(\mathbf{z}|\mathbf{x})q(c|\mathbf{x}). \tag{8}$$

Similar to VAE, we use a neural network g to model $q(\mathbf{z}|\mathbf{x})$ in VaDE:

$$\begin{aligned} [\tilde{\boldsymbol{\mu}}; \log\tilde{\boldsymbol{\sigma}}^2] &= g(\mathbf{x};\boldsymbol{\phi}) && (9)\\ q(\mathbf{z}|\mathbf{x}) &= \mathcal{N}(\mathbf{z};\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\sigma}}^2\mathbf{I}) && (10) \end{aligned}$$

where $\boldsymbol{\phi}$ is the parameter of network g.

$q(c|\mathbf{x})$ is computed as follows[1]:

$$q(c|\mathbf{x}) = E_{q(\mathbf{z}|\mathbf{x})}[p(c|\mathbf{z})] \tag{11}$$

where $p(c|\mathbf{z})$ can be computed as

$$p(c|\mathbf{z}) = \frac{p(c)p(\mathbf{z}|c)}{\sum_{c'=1}^K p(c')p(\mathbf{z}|c')} \tag{12}$$

Intuitively, in Equation 11 we want $q(c|\mathbf{x})$ to approximate $p(c|\mathbf{x})$ and use $q(\mathbf{z}|\mathbf{x})$ as a surrogate of $p(\mathbf{z}|\mathbf{x})$.

Thus, $\mathcal{L}_{\text{ELBO}}(\mathbf{x})$ in Equation 7 can be rewritten as:

$$\begin{aligned} \mathcal{L}_{\text{ELBO}}(\mathbf{x}) &= E_{q(\mathbf{z},c|\mathbf{x})}\left[\log \frac{p(\mathbf{x}, \mathbf{z}, c)}{q(\mathbf{z}, c|\mathbf{x})}\right] \\ &= E_{q(\mathbf{z},c|\mathbf{x})}[\log p(\mathbf{x}, \mathbf{z}, c) - \log q(\mathbf{z}, c|\mathbf{x})] \\ &= E_{q(\mathbf{z},c|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z}) + \log p(\mathbf{z}|c) \\ &\quad + \log p(c) - \log q(\mathbf{z}|\mathbf{x}) - \log q(c|\mathbf{x})] \end{aligned} \tag{13}$$

where all terms in Equation 13 can be substituted by Equations 4, 5, 6, 10 and 11, and the ELBO can be maximized by the SGVB estimator and the *reparameterization* trick [10] w.r.t parameters of $\{\boldsymbol{\pi}, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i, \boldsymbol{\theta}, \boldsymbol{\phi}\}, i \in \{1, \cdots, K\}$. Details of the optimization can be found in Appendix C of the supplementary materials.

Once the training is done with a maximal ELBO, a latent representation $\mathbf{z}$ can be extracted for each observed sample $\mathbf{x}$ by Equation 10, and the clustering assignments can be obtained either by Equation 12 or by treating all $\mathbf{z}$'s as samples from GMM, and applying EM-algorithm. Based on our observations, both approaches work similarly well and we pick the former for its simplicity.

[1] More details can be found in Appendix A.

3.3. Understanding the ELBO of VaDE

This section, we provide some intuitions of the ELBO of VaDE. More specifically, the ELBO in Equation 7 can be further rewritten as:

$$\mathcal{L}_{\text{ELBO}}(\mathbf{x}) = E_{q(\mathbf{z},c|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - D_{KL}(q(\mathbf{z},c|\mathbf{x})||p(\mathbf{z},c)) \quad (14)$$

The first term in Equation 14 is the *reconstruction* term, which encourages VaDE to explain the dataset well. And the second term is the Kullback-Leibler divergence from the Mixture-of-Gaussians (MoG) prior $p(\mathbf{z}, c)$ to the variational posterior $q(\mathbf{z}, c|\mathbf{x})$, which regularizes the latent embedding $\mathbf{z}$ to lie on a MoG manifold. Note that the prior $p(\mathbf{z}, c)$ here can be replaced by other mixture models [19].

To demonstrate the importance of the KL term in Equation 14, we train a VaDE model without the KL term, and use $\tilde{\boldsymbol{\mu}}$ to replace $\mathbf{z}$ when computing the *reconstruction* term of the ELBO, this is because sampling according to Equation 10 is unstable without minimizing the KL term. We refer to this model as AE+GMM since it is equivalent to training an Auto-Encoder (AE) first and applying GMM on the latent representations from the learned AE. We also show the performance of using GMM directly on the observed space (GMM), using VAE [10] on the observed space and then using GMM on the latent space from VAE (VAE+GMM), as well as the performance of LDMGI [32] and DEC [31], in Figure 2. One can see that VaDE (with KL term) outperforms AE+GMM (without KL term) significantly.

4. Experiments

In this section, we evaluate the performance of VaDE on 4 benchmarks from different modalities: MNIST [13], HHAR [28], Reuters-10K [14] and Reuters [14]. We provide extensively quantitative comparisons of VaDE with other clustering methods including GMM [2], AE+GMM, VAE+GMM [10], LDGMI [32] and the strong baseline DEC [31], as well as qualitative comparisons with Conditional GAN (CGAN) [17], GMM and DEC. We use the same network architecture as DEC for a fair comparison. The experimental results show that VaDE achieves the state-of-the-art performance on all these benchmarks. The code of VaDE is available at `http://anonymous`.

4.1. Datasets Description

The following datasets are used in our emprical experiments.

- **MNIST**: The MNIST dataset consists of 70000 handwritten digits. The images are centered and of size 28 by 28 pixels. We reshaped each image to a 784-dimensional vector.

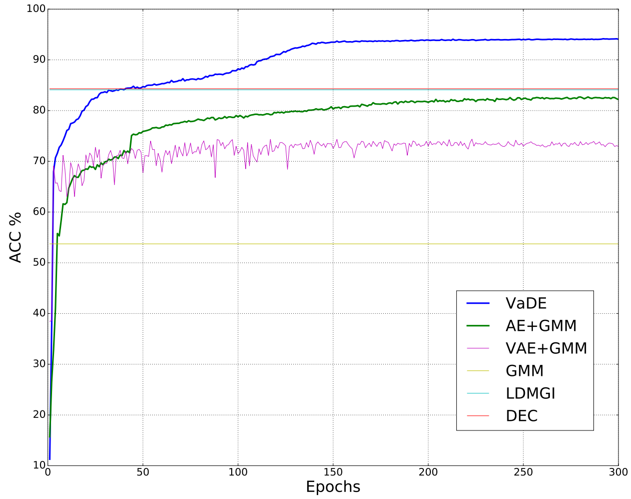


Figure 2. Clustering accuracy over number of epochs during the training on MNIST dataset for different models. *AE+GMM* is VaDE without KL divergence term in ELBO. The performances of DEC and LDMGI are taken from [31], where only the best performances are reported. We use Scikit-learn [23] to train GMM, which reports only the best performance. As a result, the curves of DEC, LDMGI, and GMM are horizontal. It is better to view the figure in color.

Dataset	Seconds Per Epoch	#Epochs
MNIST	16	150
HHAR	2	50
REUTERS-10K	1	15
REUTERS	106	4

Table 3. Training time of VaDE to achieve the performance reported in Table 2 with a single K40 GPU card.

- **HHAR**: The Heterogeneity Human Activity Recognition (HHAR) dataset [28] contains 10299 sensor records from smart phones and smart watches. All samples are partitioned into 6 categories of human activities and each sample is of 561 dimensions.

- **REUTERS**: There are around 810000 English news stories labeled with a category tree in original Reuters dataset [14]. Following DEC [31], we used 4 root categories: corporate/industrial, government/social, markets, and economics as labels and discarded all documents with multiple labels, which results in a 685071-article dataset. We computed tf-idf features on the 2000 most frequent words to represent all articles. Similar to DEC, a random subset of 10000 documents are sampled, which is referred to as Reuters-10K, since some spectral clustering methods (e.g. LDMGI [32]) cannot scale to full Reuters dataset.

Dataset	# Samples	Input Dim	# Clusters
MNIST [13]	70000	784	10
HHAR [28]	10299	561	6
REUTERS-10K [14]	10000	2000	4
REUTERS [14]	685071	2000	4

Table 1. Datasets statistics

Method	MNIST	HHAR	REUTERS-10K	REUTERS
GMM	53.73	60.34	54.72	55.81
AE+GMM	82.18	77.67	70.13	70.98
VAE+GMM	72.94	68.02	69.56	60.89
LDMGI [32]	$84.09^{\dagger}$	63.43	65.62	N/A
DEC [31]	$84.30^{\dagger}$	79.86	74.32	$75.63^{\dagger}$
VaDE	**94.06**	**84.46**	**79.83**	**77.80**

†: Taken from [31].

Table 2. Clustering accuracy (%) performance comparison on all datasets.

4.2. Experimental Setup

As mentioned before, the same network architecture as DEC is adopted by VaDE for a fair comparison. Specifically, the architectures of f and g in Equation 1 and Equation 9 are 10-2000-500-500-D and D-500-500-2000-10, respectively, where D is the input dimensionality. All layers are fully connected. Adam optimizer is used to maximize the ELBO in Equation 13, and the mini-batch size is 100. The learning rate for MNIST, HHAR and Reuters-10K is set to 0.002 and decreases every 10 epochs with a decay rate of 0.9, and the learning rate for Reuters is set to 0.0005 with a decay rate of 0.5 for every epoch. As for the generative process in Section 3.1, the multivariate Bernoulli distribution is used for MNIST dataset, and the multivariate Gaussian Distribution is used for other datasets. The number of clusters are fixed to the number of classes for each dataset, similar to DEC [31].

Although VaDE can be optimized directly with random initializations, pretraining can make the training process more stable. We use a stacked Auto-Encoder (SAE) [29] to pretrain the networks f and g. Then all data points are project into the latent space $\mathbf{z}$ by the pretrained f, where GMM is used to initialize the parameters $\{\boldsymbol{\pi}, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i\}$, where $i \in \{1, \cdots, K\}$. In practice, only a few epochs of pretraining are enough to provide a good initialization of VaDE. Since the KL term of Equation 14 can be interpreted as a regular to force the variational posterior $q(\mathbf{z}, c|\mathbf{x})$ to approximate the MoG prior, we can weight the reconstruction term and KL term of Equation 14 differently to emphasize the importance between them for better performance. In practice, the reconstruction term is weighted 5 times bigger than the KL term for HHAR dataset, and we use equal weights for the other datasets.

4.3. Quantitative Comparison

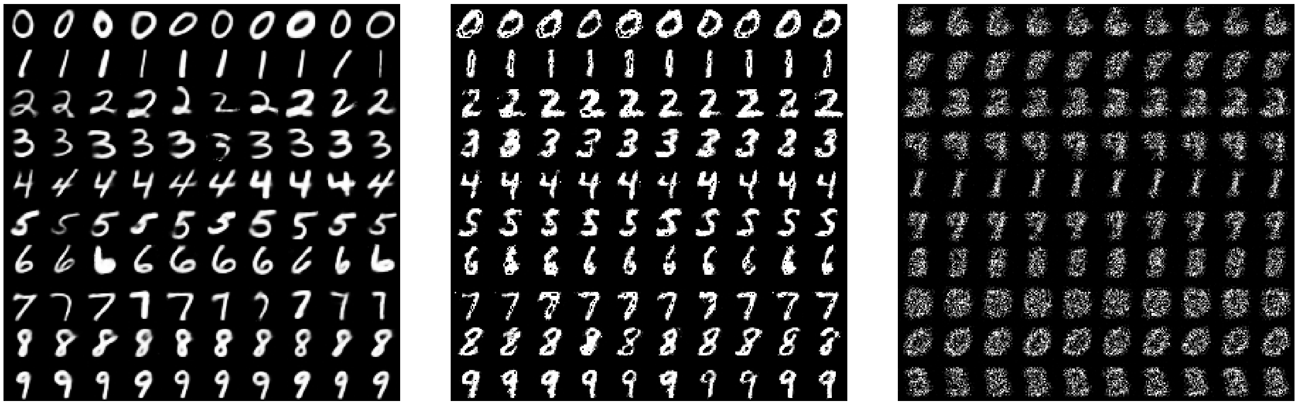
Following DEC [31], the performance of VaDE is measured by *unsupervised clustering accuracy (ACC)*, which is defined as:

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

where N is the total number of samples, l_i is the ground-truth label, c_i is the cluster assignment obtained by the model, and $\mathcal{M}$ is the set of all possible one-to-one mappings between cluster assignments and labels. The best mapping can be obtained by using the KuhnMunkres algorithm [18]. Similar to DEC, we perform 10 random restarts when initializing all clustering models and pick the result with the best objective value. As for LDMGI and DEC, we use the same configurations as their original papers. Table 2 compares the performance of VaDE with other baselines over all datasets. It can be seen that VaDE outperforms all these baselines by a large margin on all datasets. Specifically, on MNIST, HHAR, Reuters-10K and Reuters dataset, VaDE achieves ACC of 94.06%, 84.46%, 79.83% and 77.80%, which outperforms DEC with a relative increase ratio of 11.58%, 5.76%, 7.41% and 2.87%, respectively. Table 3 illustrates the training time to achieve the corresponding performances of VaDE in Table 2 on all datasets. We can see that VaDE is very efficient and is able to be optimized in reasonable time, even on the large Reuters dataset.

4.4. Generating Samples by VaDE

One major advantage of VaDE over DEC [31] is that it is by nature a *generative* clustering model and can generate highly realistic samples for any specified cluster (class). Some recent neural generative models, such as GAN [17],


(a) VaDE

(b) Conditional GAN

(c) GMM

Figure 3. The digits generated by VaDE, Conditional GAN and GMM. Digits in the same row come from the same cluster.

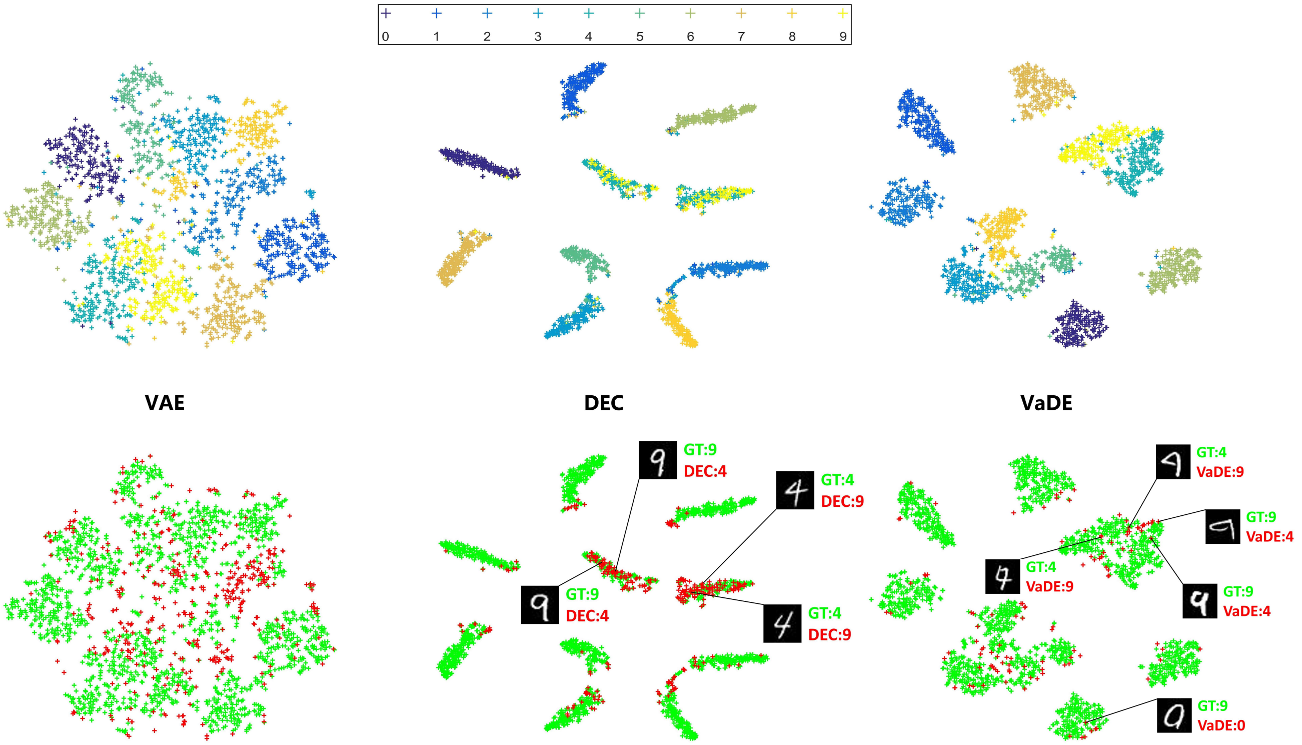
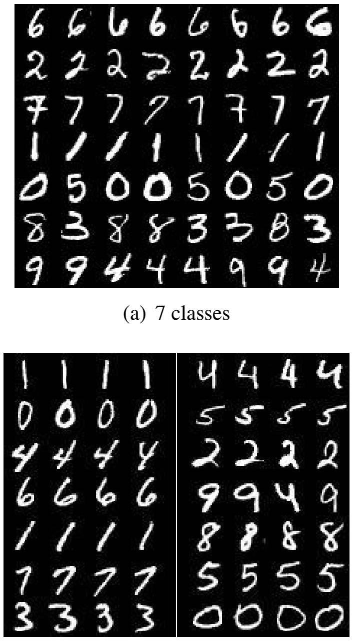


Figure 4. Visualization of the embeddings learned by VAE, DEC and VaDE on MNIST dataset, respectively. Here, GT:4 means that the ground-truth label of the digit is 4, DEC:4 means DEC clusters the digit to the cluster of 4, and VaDE:4 means the clustering result of VaDE is 4, and so on so forth for other digits. It is better to view the figure in color.

DRAW[6] and PixelRNN [22], can generate highly realistic images, but lack the capability of generating samples of a specified class.

Recently, Conditional GAN (CGAN) [17] is proposed to incorporate supervised information into GAN and thus is able to generate samples conditioned on class information. Hence, in this section, we compare the samples generated by VaDE and by CGAN with the same network architecture qualitatively. For the sake of comprehensiveness, we also generate samples from the naive GMM model conditioned on each class.

Figure 3 illustrates the generated samples for class 0 to 9 of MNIST by VaDE, CGAN and GMM, respectively. It can be seen that the digits generated by VaDE are sharper, smoother and more diverse than the alternatives. We should emphasize that, distinct from CGAN, the training procedure of VaDE does not require any label information, and is purely unsupervised.


(a) 7 classes

(b) 14 classes

Figure 5. Clustering MNIST by VaDE with different numbers of clusters. We set the number of clusters to 7 and 14, respectively, and illustrate samples belonging to each cluster by rows.

4.5. Visualization of Learned Embeddings

In this section, we visualize the learned representations of VAE, DEC and VaDE on the MNIST dataset. To this end, we use t-SNE [16] to reduce the dimensionality of the latent representation $\mathbf{z}$ from 10 to 2, and plot 2000 randomly sampled digits in Figure 4. The first row of Figure 4 illustrates the ground-truth labels for each digit, where different colors indicate different labels. The second row of Figure 4 demonstrates the clustering results, where correctly clustered samples are colored with green and incorrect ones with red.

From Figure 4 representations from VAE cannot be clustered in a distinctive way, which indicates that the single Gaussian prior of VAE is not suitable for clustering. It can also be observed that the embeddings learned by VaDE are better than those by VAE and DEC, since the number of incorrectly clustered samples is smaller. Furthermore, incorrectly clustered samples by VaDE are mostly located at the border of each cluster, where confusing samples usually appear. In contrast, a lot of the incorrectly clustered samples of DEC appear in the interior of the clusters, which confirms that the learned representations by DEC may not be suitable for clustering either. Some mistakes made by DEC and VaDE are also marked in Figure 4.

4.6. The Impact of the Number of Clusters

So far, the number of clusters for VaDE is set to the number of classes for each dataset, which is a prior knowledge. To demonstrate VaDE's representation power as an unsupervised clustering model, we deliberately choose different numbers of clusters K. Each row in Figure 5 illustrates the samples from a cluster grouped by VaDE on MNIST dataset, where K is set to 7 and 14 in Figure 5(a) and Figure 5(b), respectively. We can see that, if K is smaller than the number of classes, digits with similar appearances will be clustered together, such as 9 and 4, 3 and 8 in Figure 5(a). On the other hand, if K is larger than the number of classes, some digits will fall into sub-classes by VaDE, such as the fatter 0 and thinner 0, and the upright 1 and oblique 1 in Figure 5(b).

5. Conclusion

In this paper, we proposed Variational Deep Embedding (VaDE) which embeds the probabilistic clustering problems into a Variational Auto-Encoder (VAE) framework. VaDE models the data generative procedure by a GMM model and a neural network, and is optimized by maximizing the evidence lower bound (ELBO) of the log-likelihood of data by the SGVB estimator and the *reparameterization* trick [10]. We compared the clustering performance of VaDE with strong baselines on 4 benchmarks from different modalities, and the experimental results showed that VaDE outperforms the state-of-the-art methods by a large margin. We also showed that VaDE could generate highly realistic samples conditioned on cluster information without using supervised information during training. Note that although we use a MoG prior for VaDE in this paper, other mixture models can also be adopted in this framework flexibly.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 2

[2] C. M. Bishop, editor. *Pattern recognition and machine learning*. springer, New York, 2006. 4

[3] G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012. 2

[4] E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015. 2

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014. 2

[6] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In *ICML*, page 14621471, 2015. 2, 6

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, June 2016. 2

[8] X. He, D. Cai, Y. Shao, H. Bao, and J. Han. Laplacian regularized gaussian mixture model for data clustering. *IEEE Transactions on Knowledge and Data Engineering*, 23(9):1406–1418, 2011. 2

[9] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 2

[10] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2, 3, 4, 7

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 2

[12] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pages 2539–2547, 2015. 2

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4, 5

[14] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004. 4, 5

[15] J. Liu, D. Cai, and X. He. Gaussian mixture model with local consistency. In *AAAI*, volume 10, pages 512–517. Citeseer, 2010. 2

[16] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. 7

[17] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2, 4, 5, 6

[18] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957. 5

[19] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012. 2, 4

[20] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2002. 1

[21] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang. Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering. *IEEE Transactions on Neural Networks*, 22(11):1796–1808, 2011. 1

[22] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016. 2, 6

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 4

[24] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 2

[25] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1):19–41, 2000. 2

[26] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 22(8):888–905, 2000. 1

[27] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, page 2246, 1999. 2

[28] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 127–140. ACM, 2015. 4, 5

[29] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010. 5

[30] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007. 1

[31] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487, 2016. 1, 2, 4, 5

[32] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang. Image clustering using local discriminant models and global integration. *IEEE Transactions on Image Processing*, 19(10):2761–2773, 2010. 1, 4, 5

[33] J. Ye, Z. Zhao, and M. Wu. Discriminative k-means for clustering. In *NIPS*, pages 1649–1656, 2008. 1, 2

[34] S. X. Yu and J. Shi. Multiclass spectral clustering. In *ICCV*, pages 313–319. IEEE, 2003. 1

[35] Y. Zheng, R. S. Zemel, Y.-J. Zhang, and H. Larochelle. A neural autoregressive approach to attention-based recognition. *International Journal of Computer Vision*, 113(1):67–79, 2014. 2

[36] Y. Zheng, Y.-J. Zhang, and H. Larochelle. Topic modeling of multimodal data: An autoregressive approach. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1370–1377, June 2014. 2

[37] Y. Zheng, Y.-J. Zhang, and H. Larochelle. A deep and autoregressive approach for topic modeling of multimodal data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PP(99):1–1, 2015. 2

[38] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of International Conference on Pattern Recognition*, pages 28 – 31 Vol.2, 2004. 2