Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 19(10): 10344–10360.
DOI: 10.3934/mbe.2022484
Received: 31 May 2022
Revised: 08 July 2022
Accepted: 13 July 2022
Published: 22 July 2022

Research article

Achieving deep clustering through the use of variational autoencoders and similarity-based loss

He Ma*

College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150000, China

* **Correspondence:** Email: mahe@hrbeu.edu.cn.

Abstract: Clustering is an important and challenging research topic in many fields. Although various clustering algorithms have been developed in the past, traditional shallow clustering algorithms cannot mine the underlying structural information of the data. Recent advances have shown that deep clustering can achieve excellent performance on clustering tasks. In this work, a novel variational autoencoder-based deep clustering algorithm is proposed. It treats the Gaussian mixture model as the prior latent space and uses an additional classifier to distinguish different clusters in the latent space accurately. A similarity-based loss function is proposed consisting specifically of the cross-entropy of the predicted transition probabilities of clusters and the Wasserstein distance of the predicted posterior distributions. The new loss encourages the model to learn meaningful cluster-oriented representations to facilitate clustering tasks. The experimental results show that our method consistently achieves competitive results on various data sets.

Keywords: clustering; deep learning; data representation; variational autoencoder; network architecture

1. Introduction

Clustering is a crucial and challenging research topic in machine learning, computer vision and data mining. The goal of clustering is to separate similar and unlabeled data into several independent subsets, each representing a class. Many clustering methods [1–4] have been proposed in the literature and are widely applied to various practical problems. However, their shortcomings are becoming more and more serious for high-dimensional data. The similarity measures based on the conventional clustering method are limited by the local relationship in the data space. They thus cannot capture the hidden and hierarchical dependences of the data in the underlying structure. To solve the curse of dimensionality caused by high-dimensional data, researchers first proposed dimensionality reduction

and feature transformation methods such as principal component analysis (PCA) [5], independent component analysis [6], kernel methods [7] and non-negative matrix factorization [8]. Mapping the original high-dimensional data space to a new feature space significantly increases the density of the samples in this space and helps improve the efficiency of similarity measures. However, this feature space is not necessarily the most suitable for clustering tasks.

The rapid development of deep learning technology provides a new direction for clustering. Clustering methods with deep learning, i.e., deep clustering [9, 10], can significantly benefit clustering tasks by transforming data into cluster-oriented representations. Practical experiments show that most deep clustering methods outperform traditional clustering methods. In most deep clustering methods, high-dimensional data are mapped to a low-dimensional feature representation via a powerful deep learning algorithm; then, the clustering algorithm is directly applied. Feature representation and clustering are usually handled independently. For example [11], stacked autoencoders (AEs) are used to extract nonlinear feature representations of graphs and then directly execute the k-means algorithm. The problem with such an approach is that they do not fully exploit the potential of feature representation in deep learning. In recent years, variational deep embedding (VaDE) [12] has been proposed, which exploits the advantages of variational AEs (VAEs) to yield an unsupervised generative clustering method that combines feature representation and clustering. VAE-based frameworks can accurately capture the data distribution in the latent space to represent the observed data and learn meaningful representations. The association of feature representations and clustering algorithms can enable the learning of cluster-oriented representations.

However, existing deep clustering methods still have many challenging issues worth investigating. For example, feature representation in VaDE focuses on the global structure of the data. Still, it ignores the local structure; additionally, some recent works [13, 14] proved that similarity-based approaches in deep learning consider the local structure and thus lead to improved clustering performance.

This paper introduces a new deep clustering framework, namely, deep clustering through the use of VAEs and similarity-based loss (DVAE), that combines a VAE and a gaussian mixture model (GMM) to solve the issues above. The proposed method can optimize feature representation learning and clustering simultaneously in an end-to-end manner. Our method improves the latent space's intra-cluster compactness and inter-cluster separability during its training procedure. The main idea is to encourage the latent manifold's structure globally toward a more favorable state for clustering. The block diagram of our algorithm is shown in Figure 1. Specifically, we have added a new similarity-based graph embedding loss to the VAE's objective function. This dynamically creates a fully connected undirected graph in the latent space, treating the data as nodes on the graph and minimizing the weighted distances of their edges. We use cluster prediction to create weights for the edges and 2-Wasserstein distances of the posterior distribution over the latent space as low-dimensional vectors representing relationships. Then, we conducted many experiments to evaluate the proposed method. In addition, we further discuss the ablation experiments, parameter sensitivity analysis and visualization of our method.

The main contributions of this paper include:

- The introduction of an end-to-end VAE-based clustering framework that can simultaneously optimize embedding representations and clustering. A combination of these two aspects can promote meaningful cluster-oriented representations.
- The introduction of a novel similarity-based loss as a graph embedding extension. We use the

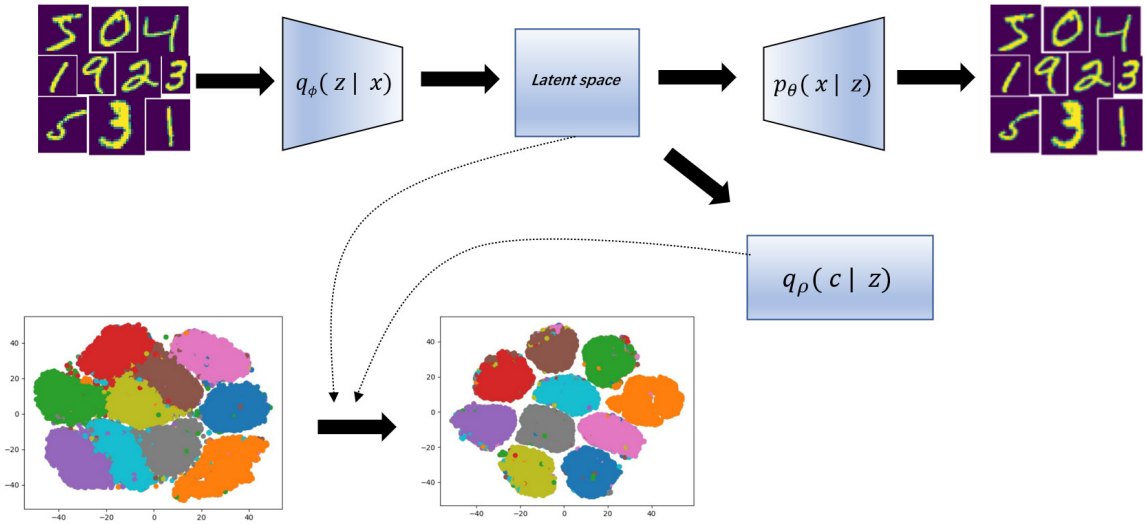


Figure 1. Schematic diagram of our proposed DVAE. $p_\theta(\mathbf{x}|\mathbf{z})$ is used for data reconstruction and $q_\phi(\mathbf{z}|\mathbf{x})$ extracts meaningful cluster-oriented feature representations from the data. During model training, the clusters in the latent space become more and more compact.

transition probabilities of cluster predictions to represent edge weights and the 2-Wasserstein distance of the posterior distribution to compute a low-dimensional representation of the relationship. The proposed loss can improve cluster-oriented representations by imposing local structural constraints, resulting in more compact clusters. To the best of our knowledge, this structure of graph embedding loss has not been applied in unsupervised clustering.

- Experiments on five publicly available data sets demonstrate that our method is highly competitive and superior to several state-of-the-art clustering methods.

The rest of this paper is structured as follows: Section 2 reviews related work. Section 3 details the framework of the VAE, including generative models, inference models and the lower bounds of evidence. At the same time, we introduce a new loss function based on the idea of a VAE. In Section 4, we conduct comprehensive experiments to evaluate the proposed method, including comparing several clustering methods and performing ablation studies, parameter sensitivity analysis and an experimental visualization. Finally, the work of this paper is summarized in Section 5.

2. Related works

2.1. Deep clustering

The goal of clustering tasks is to partition a given data set without prior knowledge and group data with similar structures or patterns into the same cluster, with different clusters having high heterogeneity. Clustering is an important and challenging research direction in machine learning; it has been studied by many scholars, resulting in many clustering algorithms. GMMs and k-means are the most widely used partitioning algorithms based on optimizing a specific cost function that allows

separation between clusters. Most traditional clustering algorithms have achieved great success on low-dimensional data. Nonetheless, they cannot achieve good clustering results in the high-dimensional space [15] when the data volume and dimensionality grow exponentially. The reason is that there is much noisy information in high-dimensional data sets, and traditional distance measures are not suitable for high-dimensional data. To solve this problem, researchers have proposed dimensionality reduction-based clustering. Clustering based on dimensionality reduction refers to mapping high-dimensional data to low-dimensional space through the use of some dimensionality reduction technique, including PCA [5], linear discriminant analysis [16] and kernel-based PCA [17]. Clustering methods based on dimensionality-reduced data have good performance on a limited number of tasks [18–20]. Furthermore, the model proposed in [21] entails the use of a spectral clustering-like approach that enables the learning of new Laplacian matrices and attainment of more robust data clustering through the use of the manifold structure and local discriminative information.

However, with the continuous improvement of practical application requirements, the traditional clustering methods based on dimensionality reduction are becoming more and more limited by the bottleneck of not being able to mine the underlying structural information of the data. Researchers have introduced deep learning to extract representations to overcome the limitations of shallow linear or nonlinear feature extraction algorithms. Deep learning allows for the mining of more complex nonlinear features and relationships. It integrates the feature extraction work into the model building process, avoids manual operations, and significantly improves the clustering effect [22]. A typical example is presented in [11], wherein the model first learns feature representations in the reconstruction task through a sparse AE-based network before k-means is applied for clustering in the feature space. AEs and traditional clustering techniques can extract more cluster-oriented features with excellent performance. The model proposed in [23] adopts a similar approach, completing representation learning through deep ternary networks. Subsequently, adversarial AE (AAE) [24] combines the discriminator from generative adversarial networks with AE, resulting in a more powerful model.

Deep embedding for clustering (DEC) [25] is a well-studied deep clustering method that uses an AE as the network framework and cluster assignment reinforcement loss as regularization. DEC ignores the problem that clustering loss cannot guarantee local structure; however, improved deep embedded clustering (IDEC) [26] improves this. IDEC uses an incomplete AE. By fusing clustering loss and the AE loss, IDEC jointly assigns cluster labels and learns features suitable for clustering and preserving the data structure. K-Autoencoders (K-DAE) [27] is an AE-based deep clustering method that extends the k-means algorithm. K-DAE uses both the reconstruction loss of an AE and the loss function of k-means to train the network.

2.2. Variational autoencoder

The VAE is a deep generative model [28]. VAEs requires the following assumptions. The low-dimensional latent variable $\mathbf{z}$ is generated from a prior distribution $p(\mathbf{z})$, for example, a multivariate standard Gaussian distribution. The high-dimensional observation data points $\mathbf{x}$ are automatically generated from the conditional distribution $p(\mathbf{x}|\mathbf{z})$ given the sampling of $\mathbf{z}$. These two structures of VAEs are called the inference model and generative model, respectively. VAEs aim to find the true posterior $p(\mathbf{x}|\mathbf{z})$ to infer the latent distribution from the observed samples. Using the

Bayes theorem, we can obtain this posterior distribution:

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})} \tag{2.1}$$

In the denominator of the above equation, the marginalization over the latent variables of the high-dimensional data distribution $p(\mathbf{x})$ is complicated. Therefore, the VAE introduces a new easy-to-solve approximate distribution $q(\mathbf{z}|\mathbf{x})$ to complete the function of the inference model and use this variational distribution to approximate the true posterior and minimize the Kullback-Leibler (KL) divergence between them, which can be written as

$$\begin{aligned} &KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})) \\ &= E_{q(\mathbf{z}|\mathbf{x})}[log(q(\mathbf{z}|\mathbf{x})) - log(p(\mathbf{x}|\mathbf{z})) - log(p(\mathbf{z})) + log(p(\mathbf{x}))] \end{aligned} \tag{2.2}$$

Since $p(\mathbf{x})$ and $\mathbf{z}$ are irrelevant, we can derive the core formula of VAE:

$$\begin{aligned} &log(p(\mathbf{x})) - KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})) \\ &= E_{q(\mathbf{z}|\mathbf{x})}[log(p(\mathbf{x}|\mathbf{z})) - kl(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] \end{aligned} \tag{2.3}$$

The term to the right of the equal sign is called the evidence lower bound (ELBO) for $\mathbf{x}$. According to Jensen's inequality [29], KL divergence is non-negative. So, $log(p(\mathbf{x}))$ is always greater than the ELBO. We can see that minimizing the distance between the variational and true distribution is equal to maximizing the ELBO. At this point, the VAE transforms the probabilistic inference problem of summing variables into the problem of optimizing the ELBO. We can write the loss of the VAE as

$$\mathcal{L}_{\text{VAE}} = -E_{q(\mathbf{z}|\mathbf{x})}[log(p(\mathbf{x}|\mathbf{z}))] + KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \tag{2.4}$$

VAEs has flexibility, theory, scalability, and robust feature extraction capabilities. Researchers have combined VAEs with classical clustering methods and succeeded in recent years. The model proposed in [30] first tries to use the mixed Gaussian distribution as the approximate posterior of the VAE. Realigning each point in the VAE latent space to its nearest class neighbor may improve the clustering effect.

The main work related to our proposed method is VaDE. VaDE and our model can learn latent feature representations based on the VAE framework and GMM priors. However, the differences are significant; specifically, our method uses a similarity-based latent space metric to preserve local data structure. On the other hand, our proposed method emphasizes that if the samples are close in the latent space, promoting shortening their distance leads to better clustering performance. Our proposed method enables the learning of more compact feature representations. The experimental results show that the proposed method outperforms VaDE.

3. Method

This section first details the clustering process for the improved VAE and its framework, including the generative model, the inference model, and the ELBO. After that, we introduce the details of the proposed similarity-based loss function.

3.1. Variational autoencoder

Let us begin with a set of D-dimensional original data points $X = \{\mathbf{x}_i \in \mathbb{R}^D\}_{i=1}^N$ derived from C nonlinear manifolds, where N is the number of data points and C is the number of corresponding classes.

3.1.1. Generative model

In our model, the observation variable $\mathbf{x}$ is generated by the latent variables $\mathbf{z}$. Instead of the single Gaussian prior, we treat the GMM as the prior latent space to capture the distribution of data. We model the distribution of $p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{c})$ which can be written as

$$\begin{aligned} p(\mathbf{c}) &= \prod_{k=1}^{C} \pi_k^{c_k} \\ p(\mathbf{z}|\mathbf{c}) &= \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_k, \mathrm{diag}(\boldsymbol{\sigma}_k^2)) \\ p_\theta(\mathbf{x}|\mathbf{z}) &= \begin{cases} Ber(\mathbf{x}|\boldsymbol{\mu}_x), & \text{if } \mathbf{x} \text{ is binary} \\ \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_x, \lambda\mathbf{I}), & \text{if } \mathbf{x} \text{ is real-valued} \end{cases} \end{aligned} \tag{3.1}$$

where $p(\mathbf{c})$ is the categorical distribution parameterized by $\pi \in \mathbb{R}_+^C$, $\sum_{k=1}^C \pi_k = 1$, $\mathbf{c} \in \{0, 1\}$, $p(\mathbf{z}|\mathbf{c})$ is the kth Gaussian component parameterized by the mean $\boldsymbol{\mu}_k$ and the variance $\boldsymbol{\sigma}_k^2$, $k = \{1, 2, ..., C\}$, $p(\mathbf{x}|\mathbf{z})$ is the Bernoulli distribution or Gaussian distribution parameterized by $\boldsymbol{\mu}_x$, $\boldsymbol{\mu}_x$ is the mean characterized by the deep neural network f with the parameters θ, which can be written as $[\boldsymbol{\mu}_x] = f(\mathbf{z}; \theta)$, λ is a predefined parameter and $\mathbf{I}$ is an identity matrix.

3.1.2. Inference model

To apply variational inference to $p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{c})$, we use the variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$ to approximate the true posterior $p(\mathbf{z}|\mathbf{x})$. We use the same strategy as VaDE, i.e., assuming that $q(\mathbf{z}, \mathbf{c}|\mathbf{x})$ is a mean-field distribution, but using a different factorization method, mathematically, follows:

$$q(\mathbf{z}, \mathbf{c}|\mathbf{x}) = q_\phi(\mathbf{z}|\mathbf{x}) q_\rho(\mathbf{c}|\mathbf{z})$$

where the mathematical forms of the two distributions to the right of the equal sign are

$$\begin{aligned} q_\phi(\mathbf{z}|\mathbf{x}) &= \mathcal{N}(\mathbf{z}|\tilde{\boldsymbol{\mu}}, \mathrm{diag}(\tilde{\boldsymbol{\sigma}}^2)) \\ q_\rho(\mathbf{c}|\mathbf{z}) &= Multinomial(\tilde{\pi}) \end{aligned} \tag{3.2}$$

where

$$\begin{aligned} [\tilde{\boldsymbol{\mu}}; log(\tilde{\boldsymbol{\sigma}}^2)] &= g(\mathbf{x}; \phi) \\ \tilde{\pi} &= h(\mathbf{z}; \rho) \end{aligned} \tag{3.3}$$

To choose a suitable $q_\rho(\mathbf{c}|\mathbf{z})$, we assume that $q(\mathbf{z}|\mathbf{x})$ is a Gaussian distribution and $q(\mathbf{c}|\mathbf{z})$ is a multinomial distribution, where $\{\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\sigma}}\}$ and $\tilde{\pi}$ are some arbitrarily determined value of Network g with the parameter ϕ and Network h with the parameter ρ.

Since the model adopts a random sampling strategy, the VAE introduces a reparameterization trick to allow backpropagation and obtain $\mathbf{z}$, as described by the following formula:

$$\boldsymbol{z}_k = \tilde{\boldsymbol{\mu}}_k + \tilde{\boldsymbol{\sigma}}_k \circ \epsilon \tag{3.4}$$

where $\circ$ denotes the element-wise product, and ϵ is sampled from the distribution $\mathcal{N}(0, \mathbf{I})$.

This paper uses the classifier h to distinguish different clusters. Once we have trained the variational encoder, we use the pre-trained model to extract features from the data and use k-means to cluster the features to obtain pseudo-labels. Since no labels are available in the clustering task, we use these pseudo labels to train the classifier.

3.1.3. Evidence lower bound

By minimizing the KL distance between $p(\mathbf{z}, \mathbf{c}|\mathbf{x})$ and $q(\mathbf{z}, \mathbf{c}|\mathbf{x})$, the loss function of the VAE with the GMM prior can be written as

$$\begin{aligned} \mathcal{L}_{\text{GVAE}}(\mathbf{x}) = &- E_{q_\phi(\mathbf{z}|\mathbf{x})}[log(p_\theta(\mathbf{x}|\mathbf{z}))] \\ &+ KL(q_\phi(\mathbf{z}|\mathbf{x})q(\mathbf{c}|\mathbf{z})||p(\mathbf{c})p(\mathbf{z}|\mathbf{c})) \end{aligned} \tag{3.5}$$

According to Eqs 3.1 and 3.2, we take the example that $\mathbf{x}$ is binary and apply the reparameterization trick; then, Eq 3.5 can be rewritten as

$$\begin{aligned} &\mathcal{L}_{\text{GVAE}}(\mathbf{x}) \\ &= -E_{q(\mathbf{z},\mathbf{c}|\mathbf{x})}[log(p(\mathbf{x}|\mathbf{z})) + log(p(\mathbf{z}|\mathbf{c})) + log(p(\mathbf{c})) \\ &\quad - log(q(\mathbf{z}|\mathbf{x})) - log(q(\mathbf{c}|\mathbf{z}))] \\ &= -\sum_{j=1}^{D} \boldsymbol{x}_i|_j log(\boldsymbol{\mu}_x|_j) - (1 - \boldsymbol{x}_i|_j) log(1 - \boldsymbol{\mu}_x|_j) \\ &\quad + \frac{1}{2}\sum_{k=1}^{C}\gamma_{ik}\sum_{m=1}^{M} log(\boldsymbol{\sigma}_k^2|_m) + \frac{\tilde{\boldsymbol{\sigma}}_i^2|_m}{\boldsymbol{\sigma}_k^2|_m} + \frac{(\tilde{\boldsymbol{\mu}}_i|_m - \boldsymbol{\mu}_k|_m)^2}{\boldsymbol{\sigma}_k^2|_m} \\ &\quad - \sum_{k=1}^{C}\gamma_{ik} log(\frac{\pi_{ik}}{\gamma_{ik}}) - \frac{1}{2}\sum_{m=1}^{M} 1 + log(\tilde{\boldsymbol{\sigma}}_i^2|_m) \end{aligned} \tag{3.6}$$

where $*|_i$ denotes the ith element of $*$. Here

$$\begin{aligned} \boldsymbol{\mu}_x|_i &= f(\mathbf{z}_i; \theta) \\ \boldsymbol{\gamma}_{ik} &= h(c = k, \mathbf{z}_i; \rho) \\ \boldsymbol{z}_i &= \tilde{\boldsymbol{\mu}}|_i + \tilde{\boldsymbol{\sigma}}|_i \circ \epsilon \\ [\tilde{\boldsymbol{\mu}}|_i, log(\tilde{\boldsymbol{\sigma}}^2|_j)] &= g(\boldsymbol{x}_i; \phi) \end{aligned} \tag{3.7}$$

where f, g and h are the generative model, inference model and classifier, respectively.

3.2. Differences from traditional variational autoencoders

Our proposed method is a generative clustering model based on VAEs. Our optimization follows steps of a VAE, using stochastic gradient variational Bayes and reparameterization techniques to optimize the ELBO. The difference between our method and the traditional VAE is reflected in two ways: 1) Our proposed method introduces a GMM to construct the model generation process. That is, $p(\boldsymbol{z})$ is a GMM instead of the normal distribution in traditional VAE; 2) Given the problem of ignoring the capture of local structures in traditional VAEs, we have added a constraint-based on the Wasserstein distance to the loss function of the VAE to help the model deal with complex data clustering problems.

3.3. Similarity-based loss

In this section, we propose a novel similarity-based loss function to preserve the local structure of the data in the latent space. Our proposed loss can be seen as an extension of graph embedding. We dynamically create a fully connected undirected graph in the latent space, treating the data as nodes and the correspondence between their distributions as edges on the graph.

We shall now discuss the construction of a weighted graph in the latent space from the samples. It is a fully connected graph with an adjacency matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$. Each non-negative element of the real symmetric matrix W represents the similarity between a pair of vertices on the graph. This matrix can be formed using various similarity measures, such as a Gaussian kernel function. Inspired by prior class information [31] in supervised learning and mixed class information [32] in semi-supervised learning, we propose a new cluster label similarity measure from latent space distance and generalize it to unsupervised study. The authors of [32] proved that, ideally, data of the same class would cluster together to form a compact cluster. Therefore, there is a positive transition probability between data points in the same cluster. We use a formula variant for transition probability to describe the extent of the data similarity; it is defined as

$$\mathbf{w}_{ij} = \sum_{k=1}^{C} \frac{\gamma_{ik}\gamma_{jk}}{\sum_{i=1}^{N}\gamma_{ik}} \tag{3.8}$$

where $\mathbf{w}_{ij}$ represents an element of $\mathbf{W}$, representing the distance between the data points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the category space.

To better characterize the similarity relationship for the data points, we propose a low-dimensional embedding relationship based on the distance of the data distribution in the low-dimensional latent space as a graph embedding. At this point, we can write out the form of the loss function in the VAE framework with graph embedding:

$$\mathcal{L}_{\text{similarity}} = -\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{w}_{ij} log\left(\frac{exp(\boldsymbol{Q}_{ij})}{\sum_{j'=1}^{N}(\boldsymbol{Q}_{ij'})}\right) \tag{3.9}$$

$$\mathbf{Q}_{ij} = (1 + d_{\text{Wasserstein}}(q(\mathbf{z}|\boldsymbol{x}_i)||q(\mathbf{z}|\boldsymbol{x}_j)))^{-1} \tag{3.10}$$

where $\mathbf{Q}_{ij}$ is a simplified notation of $\mathbf{Q}(q(\mathbf{z}|\boldsymbol{x}_i)||q(\mathbf{z}|\boldsymbol{x}_j))$, indicating the distance between the two distributions. $d_{\text{Wasserstein}}$ is a variant of the 2-Wasserstein distance. Since $q(\mathbf{z}|\boldsymbol{x})$ is in Gaussian form, there is a closed-form solution here, given by

$$d_{\text{Wasserstein}}(q(\mathbf{z}|\boldsymbol{x}_i)||q(\mathbf{z}|\boldsymbol{x}_j)) = ||\tilde{\boldsymbol{\mu}}_i - \tilde{\boldsymbol{\mu}}_j||_2^2 + Tr(\Sigma_1 + \Sigma_2 - 2(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}) \tag{3.11}$$

Given Eq 3.11, we can further simplify the third term on the right side of the equal sign as

$$\begin{aligned}\Sigma_1 + \Sigma_2 - 2(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2} &= \text{diag}(\tilde{\boldsymbol{\sigma}}_i^2) + \text{diag}(\tilde{\boldsymbol{\sigma}}_j^2) - 2\text{diag}(\tilde{\boldsymbol{\sigma}}_i\tilde{\boldsymbol{\sigma}}_j)\\ &= \text{diag}(\tilde{\boldsymbol{\sigma}}_i^2 + \tilde{\boldsymbol{\sigma}}_j^2 - 2\tilde{\boldsymbol{\sigma}}_i\tilde{\boldsymbol{\sigma}}_j)\\ &= \text{diag}((\tilde{\boldsymbol{\sigma}}_i - \tilde{\boldsymbol{\sigma}}_j)^2)\end{aligned} \tag{3.12}$$

Substituting Eq 3.12 into Eq 3.11, we get

$$d_{\text{Wasserstein}}(q(\mathbf{z}|\boldsymbol{x}_i)||q(\mathbf{z}|\boldsymbol{x}_j)) = ||\tilde{\boldsymbol{\mu}}_i - \tilde{\boldsymbol{\mu}}_j||_2^2 + ||\tilde{\boldsymbol{\sigma}}_i - \tilde{\boldsymbol{\sigma}}_j||_2^2 \tag{3.13}$$

Unlike existing similarity-based methods, which consider the similarity of the original data in the Euclidean space, our proposed loss function focuses on the relationship between the posterior prediction of the data and the similarity in the latent space. Similarity increases when data in the latent space is close, and vice versa. This loss function encourages data in the same cluster to cluster together in the latent space, and data in different clusters are mutually exclusive to promote separation.

3.4. Variational autoencoder with latent clusters

We can write the final loss as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{GVAE}} + \beta \mathcal{L}_{\text{similarity}} \tag{3.14}$$

where $\beta \geq 0$ is a weight coefficient that controls the proposed loss function. For our proposed loss to always have a positive effect on the model during the learning phase, we have the following settings: the weight β is 0 for the first T iterations and takes appropriate values thereafter. It works for all tasks, and the model performs best when T is 50.

4. Experiments

The experiment proves the superiority of our proposed method. To make a fair comparison, we use the same network structure and parameters as VaDE. Like VaDE, we stack the AE into a deep VAE and then pre-train the generative and inference models to eliminate the initial VAE's problems. We use k-means clustering to complete the initialization of the classifier and Gaussian model by generating pseudo-labels in the latent space clustering to train the classifier network and determining the initialization of the mean and variance of the GMM. The pseudo-code of our algorithm is as follows:

Algorithm 1: DVAE

Input: samples $X = \{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=1}^N$, networks

initialization

for *each minibatch* $\{\boldsymbol{x}_i\}_{i=1}^M$ *in* X **do**

  compute $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\tilde{\mu}, \text{diag}(\tilde{\boldsymbol{\sigma}}^2))$

  sample $\boldsymbol{z}^{(i)} \sim q_\phi(\mathbf{z}|\mathbf{x})$

  compute $q_\phi(\mathbf{z}|\boldsymbol{x}_i)$ and $q_\phi(\mathbf{z}|\boldsymbol{x}_j)$

  compute $p_\theta(\mathbf{x}|\mathbf{z}) = Ber(\mathbf{x}|\boldsymbol{\mu}_x)$ or $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_x, \lambda\mathbf{I})$

  generate $\boldsymbol{x}^{(i)} \sim p_\theta(\mathbf{x}|\mathbf{z})$

  evaluate the objective function $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{GVAE}} + \beta\mathcal{L}_{\text{similarity}}$

  update parameters

end

4.1. Data sets

The data sets used in this experiment include MNIST, HHAR, REUTERS-10K and USPS. Specifically,

- MNIST is a database containing handwritten digital images from 250 different people. It includes

Table 1. Dataset statistics.

Dataset	Sample No.	Input Dim	Cluster
MNSIT	70000	784	10
HHAR	10200	561	6
REUTERS-10K	10000	2000	4
STL-10	13000	2048	10
USPS	9298	256	10

a training set of 60,000 examples and a test set of 10,000 samples. MNIST consists of 10 types of handwritten digital pictures and digital labels composed of 28×28 pixels.

- HHAR is a heterogeneous human activity recognition data set acquired from smartphones and smartwatches, and it contains 10,200 sensor records. The number of dimensions recorded by the sensor is 561 dimensions, which are divided into six categories of human activities, including Biking, Sitting, Standing, Walking, Stair Up and Stair down.
- Reuters is an English text data set from 11,228 press articles in Reuters, labeled with 46 topics. Reuters-10k [33] is a subset of 10,000 samples randomly sampled from Reuters, and it includes four categories: corporate/industrial, government/social, markets and economics.
- STL-10 is a database of 13,000 labeled and 100,000 unlabeled color images. Compared to MNIST, it has a higher resolution (96×96 pixels) and is highly complex (10 real-world classes). We applied ResNet-50 to reduce the image dimensions to 2048.
- USPS is a data set containing 9,298 digital images from envelopes automatically scanned by the United States Postal Service. It consists of 16×16 pixel grayscale samples, and each image in the data set has been centered and normalized.

The detailed information is shown in Table 1.

4.2. Experimental settings

For the proposed method, we use the same network structure as VaDE, with $D-500-500-2000-10$ as the encoder network structure, $10-2000-500-500-D$ as the decoder and $10-C(C=10)$ or $10-C-C(C<10)$ for the classifier, where D represents the number of dimensions of the data and C represents the number of classes. All layers are fully connected in our model, and ReLU is used as an activation function. We use a batch size of 300 for all data sets. MNIST, HHAR, REUTERS-10K, STL-10 and USPS have an initial learning rate of 0.002, reduced every 10 epochs with a decay rate of 0.9. During the first 50 epochs, the value of β is 0.

4.3. Evaluation metrics

We measured the performances of our proposed method and the benchmarks using clustering accuracy (ACC) and normalized mutual information (NMI).

ACC is defined as

$$ACC = \max_{m} \frac{\sum_{n}^{N} \mathbf{1}\{l_n = m(c_n)\}}{N} \tag{4.1}$$

where l_n and c_n represent the true label and cluster assignment generated by the input x_n, respectively.

m covers all possible one-to-one mappings between the predicted labels produced by the clustering and ground-truth labels.

NMI is defined as

$$NMI = \max_{m} \frac{I(l;c)}{\max\{H(l), H(c)\}} \tag{4.2}$$

where $I(\cdot;\cdot)$ and $H(\cdot)$ represent the mutual information and the entropy, respectively.

4.4. Comparison with several clustering approaches

We compared several methods to verify the effectiveness of our proposed method, including traditional clustering methods, i.e., a GMM and k-means algorithm, and deep clustering methods, i.e., AE+GMM and K-DAE [27], IDEC [26] and VaDE algorithms [12].

We present the ACC and NMI results obtained for each method in Tables 2 and 3 and highlight the best scores. From these two tables, we have the following observations. 1) Deep clustering performs better than traditional clustering. The excellent performance of the GMM and k-means algorithm on STL10 is attributable to the data being processed by ResNet-50, which is equivalent to the effect of applying traditional clustering after deep learning. 2) The VAE was is more stable and effective than the AE on these five data sets. On MNIST, HHAR, Reuters-10K and STL-10, VaDE demonstrated relatively competitive performance compared with other existing methods. 3) Our proposed method achieved the highest ACC for all data sets. On USPS, in particular, it far outperformed any clustering method.

Table 2. ACC results for various clustering methods.

Method	MNIST	HHAR	REUTERS-10K	STL-10	USPS
GMM	47.80%	55.65%	55.47%	83.91%	58.70%
K-means	53.23%	60.24%	54.04%	83.87%	67.32%
AE+GMM	79.34%	60.72%	70.13%	79.83%	84.16%
IDEC	77.24%	60.72%	58.80%	50.32%	75.90%
K-DAE	77%	47%	76%	61%	76%
VaDE	83.51%	78.72%	79.83%	85.73%	73.14%
DVAE	**95.53%**	**85.27%**	**82.80%**	**86.45%**	**93.15%**

Table 3. NMI results for various clustering methods.

Method	MNIST	HHAR	REUTERS-10K	STL-10	USPS
GMM	38.24%	62.12%	38.85%	74.81%	53.59%
K-means	49.97%	58.87%	41.28%	74.77%	61.49%
AE+GMM	81.41%	48.24%	28.23%	42.54%	78.45%
IDEC	76.34%	69.88%	38.29%	45.69%	80.50%
K-DAE	83%	69%	50%	65%	78%
VaDE	81.99%	68.84%	53.43%	75.80%	70.20%
DVAE	**89.51%**	**78.19%**	**57.59%**	**77.88%**	**86.67%**

4.5. Ablation study

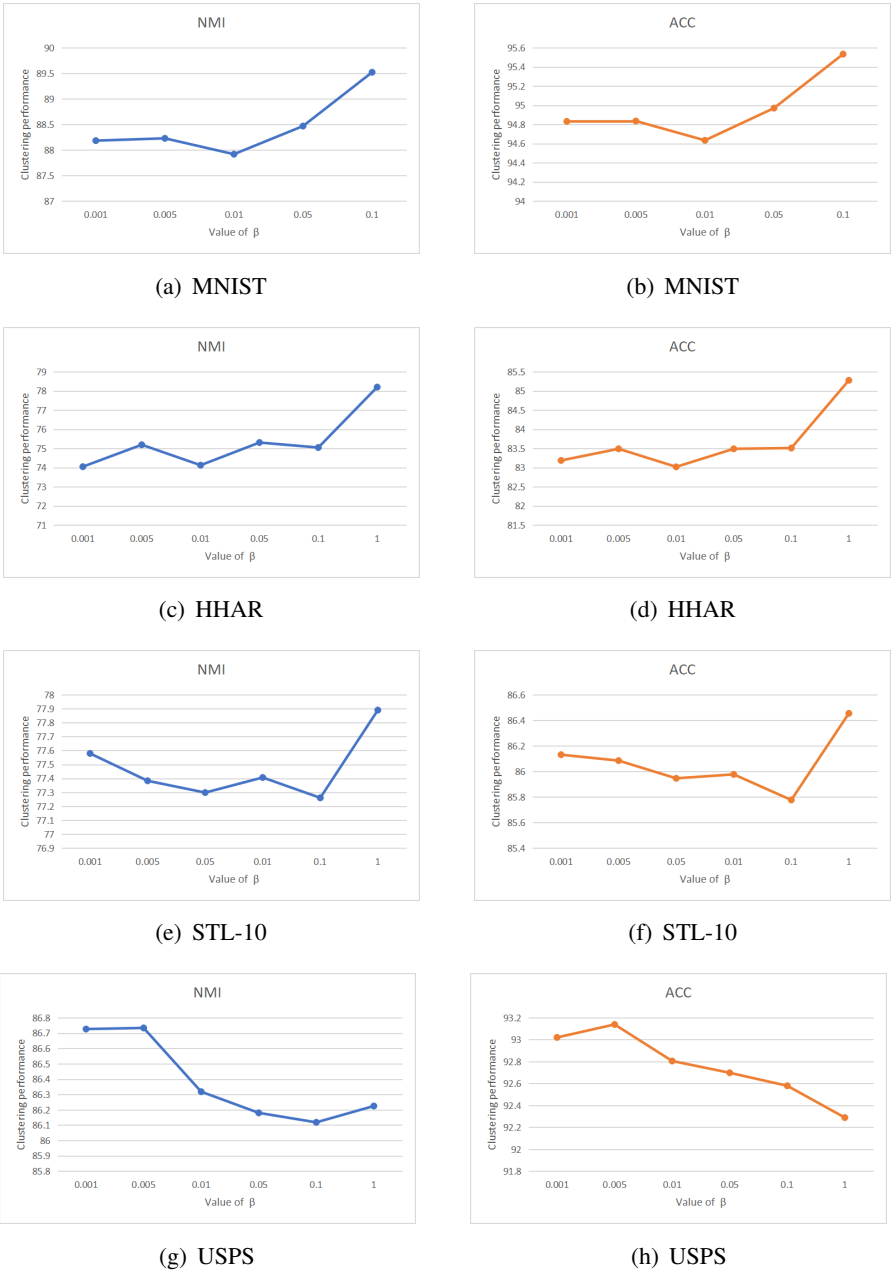
To verify the effectiveness of the similarity-based loss term introduced in our method, we further conducted ablation studies on four data sets, i.e., HHAR, Reuters-10k, STL-10K and USPS. Specifically, the network is learned by removing the similarity-based loss term and studying the network's performance. The experimental results are shown in Table 4, from which we can make the following observations. We can see that adding similarity-based terms benefits the clustering task. On HHAR and STL-10, the clustering performance improved by 2.48% and 0.55%, respectively, and the term of similarity was relatively mild for STL-10, although there was still a slight improvement. On Reuters-10k and USPS, adding similarity items significantly affected the clustering effect. The clustering performance improved by 5.33% on Reuters-10k and by an astonishing 45.71% on USPS, which strongly proves the effect of the similarity-based loss term.

Table 4. Ablation studies

Method	HHAR	REUTERS-10K	STL-10	USPS
DVAE (no similarity-based term)	82.79%	77.47%	85.96%	47.44%
DVAE	85.27%	82.80%	86.45%	93.15%

4.6. Parameter sensitivity

The parameter β affects the contribution of our proposed similarity term to the total loss. To choose appropriate weights, we took {0.001, 0.005, 0.01, 0.05, 0.1, 1} and report the effect of β on the NMI and ACC for the four data sets in Figure 2, respectively. The experiments show that taking the highest β value in most cases will yield the best ACC and NMI values. For example, in the cases of HHAR and STL-10, $\beta = 1$ achieved the best results. In the case of MNIST, $\beta = 0.1$ achieved the best results because DVAE suffers from the same problem as VaDE when $\beta = 1$ gets stuck in undesired local minima or saddle points. The best way at the moment is to use pre-training to overcome that problem. At the same time, in the case of USPS, although β shows a downward trend on the graph, the magnitude is minimal. To sum up, we recommend that β takes the biggest value that the model can sustain.



(a) MNIST (b) MNIST

(c) HHAR (d) HHAR

(e) STL-10 (f) STL-10

(g) USPS (h) USPS

Figure 2. Effects of different β values on clustering performance. The β range is {0.001, 0.005, 0.01, 0.05, 0.1, 1}.

4.7. Experimental visualization

We made the following two visualizations. First, we specifically compared the performances of the DVAE and VaDE on the MNIST data sets; as shown in Figure 3, the abscissa is the number of iterations, and the ordinate is the value of ACC. As the number of iterations increased, the DVAE consistently outperformed VaDE. Second, we used the t-SNE to visualize the output of the encoder of the VAE to show the degree of separation of the clusters in the latent space in Figure 4. As shown in

the figure, as the number of iterations increased, clusters of the same class eventually became more compact.

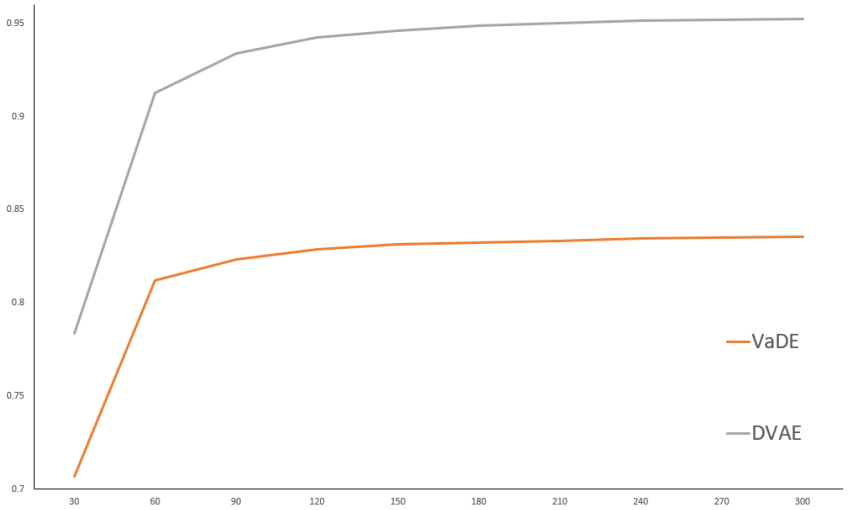
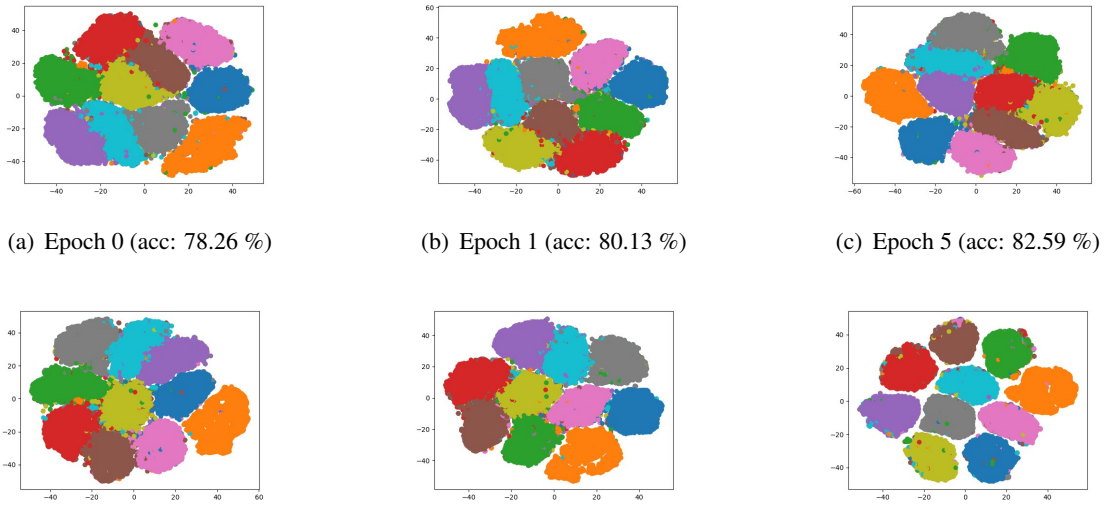


Figure 3. Comparison of ACC results for VaDE and the DVAE for the MNIST data set.



(a) Epoch 0 (acc: 78.26 %) (b) Epoch 1 (acc: 80.13 %) (c) Epoch 5 (acc: 82.59 %)

(d) Epoch 50 (acc: 92.07 %) (e) Epoch 120 (acc: 92.75 %) (f) Epoch end (acc: 96.11 %)

Figure 4. Visualization of the MNIST data set. The figure shows the clusters in the latent space during the training of the proposed method. Different colors represent different categories. The ACC during training is shown in parentheses.

5. Conclusions

We have proposed a new loss function for unsupervised clustering. This loss function enhances clustering performance by encouraging the global structure of the data manifold in the latent space toward a more compact direction. Specifically, this loss function is the cross-entropy of the predicted cluster transition probabilities and the Wasserstein distance of the predicted posterior distribution. We evaluated our method on four widely used data sets, demonstrating that our method is more robust and effective.

Conflict of interest

The authors declared that they have no conflicts of interest to this work.

References

1. A. K. Jain, Data clustering: 50 years beyond k-means, *Pattern Recognit. Lett.*, **31** (2009), 651–666. https://doi.org/10.1016/j.patrec.2009.09.011
2. A. K. Jain, R. C. Dubes, *Algorithms for clustering data*, Prentice-Hall, University of Michigan, 1998.
3. A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: A review, *ACM Comput. Surv.*, **3** (1999), 264–323. https://doi.org/10.1145/331499.331504
4. R. Xu, D. C. Wunsch II, Survey of clustering algorithms, *IEEE Trans. Neural Netw. Learn. Syst.*, **3** (2005), 645–678. https://doi.org/10.1109/TNN.2005.845141
5. K. Pearson, On lines and planes of closest fit to systems of point in space, *Philos. Mag.*, **2** (1901), 559–572.
6. P. Comon, Independent component analysis, A new concept?, *Signal Process.*, **36** (1994), 287–314. https://doi.org/10.1016/0165-1684(94)90029-9
7. V. N. Vapnik, Statistical learning theory, in *Learning from Data: Concepts, Theory, and Methods*, Wiley Interscience, (2007), 99–150.
8. C. Zheng, D. Huang, L. Zhang, X. Kong, Tumor clustering using nonnegative matrix factorization with gene selection, *IEEE T. Inf. Technol. B.*, **13** (2009), 599–607. https://doi.org/10.1109/TITB.2009.2018115
9. J. Wang, J. Jiang, Unsupervised deep clustering via adaptive GMM modeling and optimization, *Neurocomputing*, **433** (2021), 199–211. https://doi.org/10.1016/j.neucom.2020.12.082
10. J. Cai, S. Wang, C. Xu, W. Guo, Unsupervised deep clustering via contractive feature representation and focal loss, *Pattern Recogn.*, **123** (2022). https://doi.org/10.1016/j.patcog.2021.108386
11. F. Tian, B. Gao, Q. Cui, E. Chen, T. Liu, Learning deep representations for graph clustering, *AAAI Conf. Artif. Intell.*, (2014), 1293–1299. https://doi.org/10.1609/aaai.v28i1.8916

12. Z. Jiang, Y. Zheng, H. Tan, B. Tang, H. Zhou, Variational deep embedding: an unsupervised and generative approach to clustering, *Int. Joint Conf. Artif. Intell.*, (2017), 1965–1972. https://doi.org/10.48550/arXiv.1611.05148
13. L. Yang, N. Cheung, J. Li, J. Fang, Deep clustering by Gaussian mixture variational autoencoders with graph embedding, *IEEE Int. Conf. Comput. Vis.*, (2019), 6449-6458. https://doi.org/10.1109/ICCV.2019.00654
14. V. Prasad, D. Das, B. Bhowmick, Variational clustering: Leveraging variational autoencoders for image clustering, *Int. Jt. Conf. Neural Networks*, (2020). https://doi.org/10.1109/IJCNN48605.2020.9207523
15. C. C. Aggarwal, C. K. Reddy, Data clustering: algorithms and applications, in *Data Mining and Knowledge Discovery Series*, CRC Press, (2013).
16. M. Li, B. Yuan, 2D-LDA: A statistical linear discriminant analysis for image matrix, *Pattern Recognit. Lett.*, **26** (2005), 527–532. https://doi.org/10.1016/j.patrec.2004.09.007
17. B. Schölkopf, A. Smola, K. Müller, Nonlinear component analysis as a kernel eigenvalue problem, *Neural Comput.*, **10** (1998), 1299–1319. https://doi.org/10.1162/089976698300017467
18. X. Chen, D. Cai, Large scale spectral clustering with landmark-based representation, *AAAI Conf. Artif. Intell.*, (2011), 313–318. https://doi.org/10.1109/TCYB.2014.2358564
19. D. Cai, X. He, X. Wang, H. Bao, J. Han, Locality preserving nonnegative matrix factorization, *Int. Joint Conf. Artif. Intell.*, (2009).
20. G. Trigeorgis, K. Bousmalis, S. Zafeiriou, B. W. Schuller, A deep semi-nmf model for learning hidden representations, *Int. Conf. Mach. Learn.*, **32** (2014), 1692–1700. https://doi.org/10.5555/3044805.3045081
21. Y. Yang, D. Xu, F. Nie, S. Yan, Y. Zhuang, Image clustering using local discriminant models and global integration, *IEEE Trans. Image Process.*, (2010). https://doi.org/10.1109/TIP.2010.2049235
22. J. Wang, A. Hilton, J. Jiang, Spectral analysis network for deep representation learning and image clustering, *Int. Conf. Multimedia Expo*, (2019), 1540–1545. https://doi.org/10.1109/ICME.2019.00266
23. O. Nina, J. Moody, C. Milligan, A decoder-free approach for unsupervised clustering and manifold learning with random triplet mining, *IEEE Int. Conf. Comput. Vis. Workshops*, (2019). https://doi.org/10.1109/ICCVW.2019.00493
24. S. A. Makhzani, J. Shlens, N. Jaitly, I. J. Goodfellow, Adversarial autoencoders, preprint, arXiv: 1511.05644.
25. J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, *Int. Conf. Mach. Learn.*, (2016), 740–749. https://doi.org/10.48550/arXiv.1511.06335
26. X. Guo, L. Gao, X. Liu, J. Yin, Improved deep embedded clustering with local structure preservation, *Int. Conf. Mach. Learn.*, (2017), 1753–1759.
27. Y. Opochinsky, S. E. Chazan, S. Gannot, J. Goldberger, K-Autoencoders deep clustering, *IEEE trans. acoust. speech signal process.*, (2020), 4037–4041. https://doi.org/10.1109/ICASSP40776.2020.9053109

28. D. P. Kingma, M. Welling, Auto-encoding variational Bayes, preprint, arXiv: 1312.6114.

29. J. L. W. V. Jensen, Sur les fonctions convexes et les inégalités entre les valeurs moyennes, *Acta Math.*, **30** (1906), 175–193. https://doi.org/10.1007/BF02418571

30. N. Dilokthanakul, P. A. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran, et al., Deep unsupervised clustering with gaussian mixture variational autoencoders, preprint, arXiv: 1611.02648.

31. A. M. Martinez, A. C. Kak, PCA versus LDA, *IEEE Trans. Pattern Anal. Mach. Intell.*, **23** (2001), 228–233. https://doi.org/10.1109/34.908974

32. K. Kamnitsas, D. C. Castro, L. L. Folgoc, I. Walker, R. Tanno, D. Rueckert, et al., Semi-supervised learning via compact latent space clustering, *Int. Conf. Mach. Learn.*, (2018), 2459–2468. https://doi.org/10.48550/arXiv.1806.02679

33. B. Yang, X. Fu, N. D. Sidiropoulos, M. Hong, Towards k-means-friendly spaces: Simultaneous deep learning and clustering, *Int. Conf. Mach. Learn.*, (2017), 5888–5901.

AIMS Press