Multimedia Tools and Applications (2022) 81:34309–34324
https://doi.org/10.1007/s11042-021-11597-2

1168: DEEP PATTERN DISCOVERY FOR BIG MULTIMEDIA DATA

Deep boundary-aware clustering by jointly optimizing unsupervised representation learning

Ru Wang[1] · Lin Li[2] · Peipei Wang[2] · Xiaohui Tao[3] · Peiyu Liu[1]

Received: 14 September 2020 / Revised: 8 April 2021 / Accepted: 22 September 2021 /
Published online: 24 January 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Deep clustering obtains feature representation generally and then performs clustering for high dimension real-world data. However, conventional solutions are two-stage embedding learning-based methods and these two processes are separate and independent, which often leads to clustering results cannot feedback to optimize the representation learning and reduces the performance of deep clustering. In this paper, we aim to propose a deep boundary-aware clustering by jointly optimizing unsupervised representation learning. More specifically, we joint boundary-aware variational auto-encoder and deep regularized clustering for deep regularized clustering for unsupervised learning, named **B**oundary-**a**ware **DE**ep **C**lustering (BaDEC). BaDEC is able to learn feature representation and clustering simultaneously, and it introduces deep regularized clustering to reduce the unreliability of the similarity measures. In particular, we present a boundary-aware variational auto-encoder that tunes variable evidence lower bounds flexibly to assist feature representation learning better for more accurate clustering. Extensive experiments on various datasets from multiple domains demonstrate that the proposed method outperforms several popular comparison baseline methods.

Keywords Unsupervised representation learning · Deep clustering · Variational bounds

✉ Lin Li
cathylilin@whut.edu.cn

✉ Peiyu Liu
liupy@sdnu.edu.cn

Peipei Wang
ppwang07@whut.edu.cn

Xiaohui Tao
Xiaohui.Tao@usq.edu.au

[1] School of Information Science and Engineering, Shandong Normal University, Jinan, China

[2] School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China

[3] School of Sciences, University of Southern Queensland, Toowoomba, Australia

1 Introduction

Clustering is one of the fundamental research procedures of data mining and machine learning, and it has been successfully applied to a large variety of tasks [1, 13, 22, 29], such as text clustering, speech separation, image retrieval and clustering-based wireless networks application [28]. Well-known approaches include hierarchical clustering [30], partition clustering [27], spectral clustering [17], density clustering [25], and fuzzy clustering [15]. Despite these standard clustering algorithms have progressed, it is adversely affected when coping with high-dimensional data. This is called the curse of dimensionality. To tackle this problem, a common approach is to project high-dimensional data to low-dimensional manifold.

In recent years, deep learning has achieved widespread success in numerous machine learning tasks [9–11, 14, 24] which reduces dimension by embedding data into a lower dimensional space and provides deep reliable representation for downstream tasks. Deep clustering integrates unsupervised clustering and deep neural networks, it aims to learn the deep representation in an unsupervised learning way, and then clusters the embedded data in the new subspace. There are many studies joint deep neural networks and clustering object optimization methods to complete unsupervised embedding learning and clustering task [5, 7, 16, 20, 26]. Although these works have achieved good performance, we argue that they still suffer from inherent limitations. First, most of them are two-stage embedding learning-based methods (see Fig. 1) and feature representation is not able to be updated by clustering feedback. Second, some studies adopt similarity metrics between high-dimensional data to estimate the differences between samples, leading to suboptimal clustering results. Third, existing works pay more attention to maximize evidence lower bounds and obtain tighter variational bounds in the Variational Auto-Encoder (VAE) [12] framework. In fact, tighter variational bounds are not necessarily better [18], and deep clustering algorithms that require tuning the variational bounds flexibly to fit feature learning.

To address the mentioned challenging limitations, in this paper, we propose a deep boundary-aware clustering by jointly optimizing unsupervised representation learning (BaDEC). Specifically, the proposed BaDEC is able to learn the feature representation

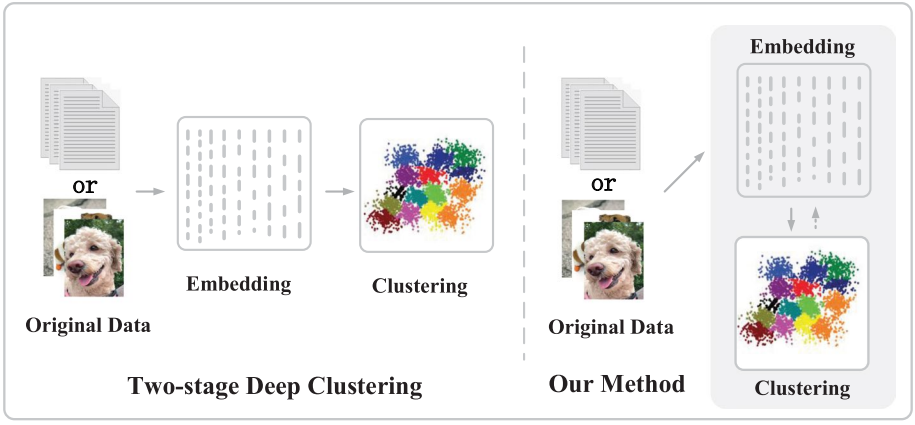

Fig. 1 The difference between two-stage embedding learning methods and our method

and performs clustering simultaneously in a unified framework. As shown in Fig. 1, we bridge embedding and clustering synergistically, enabling the high-confidently feature representation. Besides, to avoid the unreliability of similarity metrics, we explore a deep regularized clustering to gain more accurate prediction results. In particular, we design a boundary-aware variational auto-encoder to achieve the flexibility of variable evidence lower bounds (ELBOs) for deep clustering. Finally, extensive experimental results show that our BaDEC achieves better performance compared with baseline methods in clustering predication on various datasets. The main contributions of this paper can be summarized as follows.

- We propose a deep boundary-aware clustering by jointly optimizing unsupervised representation learning BaDEC, which combines boundary-aware variational auto-encoder and deep regularized clustering. In our BaDEC, we could learn feature representation and clustering simultaneously. Instead of the similarity metric, we estimate the differences between samples by exploiting an deep regularized clustering.
- To the best of our knowledge, this is the first work to focus on boundary-aware variational auto-encoder for deep clustering. More specifically, we define a regulable optimization objective that tunes variable evidence lower bounds flexibly to assist feature representation learning better, resulting in more accurate clustering prediction.
- We conduct comprehensive experiments to evaluate the effectiveness of our BaDEC. Experimental results demonstrate that the proposed method performs better than comparison baseline methods on six benchmark datasets for deep clustering task.

2 Related work

Deep clustering has been extensively studied in data analysis in terms of deep feature representation and clustering [5, 7, 11, 19, 24]. These methods generally combine deep learning and clustering algorithms to deal with high-dimensional data. Since the effective deep representation of deep neural networks, it provides a high-quality feature space for the clustering algorithms. Meanwhile, clustering algorithms can estimate better the difference between samples by the low-dimensional features of samples based on deep feature representation. Deep neural network based framework contributes to joint optimization for various tasks [4].

Deep clustering Recently, deep embedded clustering(DEC) [24] was proposed to learn feature representations and cluster assignments using deep neural networks. Despite DEC performs well, DEC defined the similarity between embedded feature and centroid based on stacked denoising auto-encoder [21]. As a result, the feature embedding space is not suitable for clustering task. IDEC [7] is an improved deep clustering method that can jointly optimize cluster labels assignment and learn features. SDEC [19] is a semi-supervised deep embedded clustering method to learn feature representations and perform clustering task simultaneously. VaDE [11] is an unsupervised and generative approach to clustering that combines gaussian mixture model (GMM) and variational auto-encoder (VAE). However, these works are overly dependent on the similarity calculation between samples. To avoid this issues, DEPICT [5] employs a multinomial logistic regression function to predict the probabilistic cluster assignments and it built an adversarial mechanism for better feature representation learning.

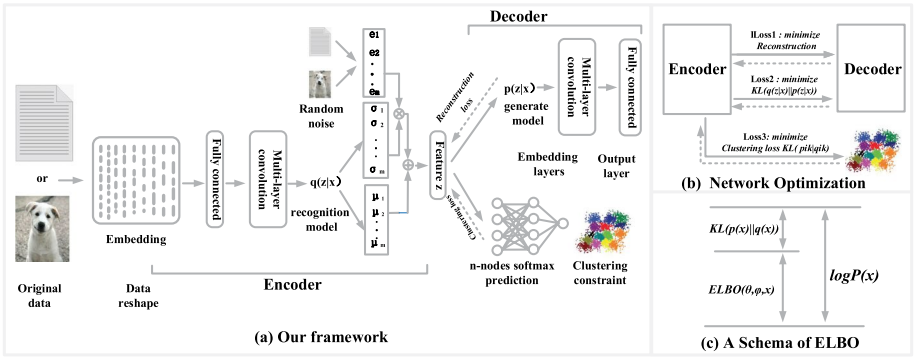

Fig. 2 The architecture of the proposed BaDEC. **a** The Encoder is a recognition model $q(z|x)$ to obtain the feature distribution of the implicit features from the multi-layer convolutional layer's output. The Decoder is a generate model $p(z|x)$ to reconstruct the sample. Deep regularized clustering captures the results from the implicit feature space and construct clustering constraints. **b** Network optimization of BaDEC. **c** A schema of ELBO

The tighter evidence lower bounds are not necessarily better Variational auto-encoder architecture is one of the most important methods used in deep clustering methods. In a general way, variational bounds are utilized to train autoencoder fameworks [2, 6, 8] by minimizing the KL divergence of P and Q distributions. As following formulas, we present a general formula of $KL(P||Q)$, which show Fig. 2(c).

$$KL(P||Q) = -L(\theta, \varphi, x) + \log P(x),$$

$$L \to \max \Rightarrow KL(P||Q) \to \min,$$

where $\log P(x)$can be regarded as a constant, thus, minimizing the $KL(P||Q)$ is to maximize the L, and the L is well known as ELBOs. Such as [6] uses the variational auto-encoder structure to complete the depth generation model to realize the latent variable image modeling by optimizing the standard variational evidence lower bound. Besides, DEC [24], SDEC [19], and VaDE [11] employ standard evidence lower bounds, IDEC [7] adopts a tighter bound to optimize clustering tasks. Deep clustering is a two-task learning, feature learning and clustering. This method minimizes a lower bound of variational evidence to complete the task, which can be shown in the Fig. 2(c). Compared with the standard boundary and the tightened boundary, the performance of various methods is uneven.

In a recent work, [18] develops three importance weighted auto-encoder methods for the tight boundary problem, which demonstrates that tighter bounds are not necessarily better optimization tasks. However, in [18] they merely considers the terms of signal and noise generation of different tightness boundaries, and regardless of the helpful in feature representation leaning framework. To better optimizing our method, we target the variable boundaries from the view of feature learning and feature reconstruction, and how to integrate variable boundaries to design a better optimization objective. As mentioned before, one limitation of this work is that they maximize evidence lower bounds and obtain tighter variational bounds. As a matter of fact, it requires tuning the variational bounds flexibly to

fit feature learning for clustering task. Distinct from previous works, inspired by [18], we propose a joint optimization objective that can obtain boundary-aware ELBOs in a adjustable way for better feature learning. Besides, our BaDEC combines the embedding and clustering jointly and feature representaion can be updated in terms of clustering results. As a specific component, our design-well automated clustering can capture automatically the difference of samples instead of using similarity calculation.

3 Our proposed BaDEC

3.1 Notation

Consider the clustering task is to divide N samples into K clusters. A set of samples X=[x_1, x_2, ... , x_n] ∈ R^{dxn}, we define the recognition model q(z|x) as a probabilistic encoder, and the general model p(z|x) as a generate decoder by z. Based on this, we define q(z_i | x_i) as the sample distribution of x_i, and p(z_i | x_i) denotes distribution from z_i sampling. q(z_i | x_i) and p(z_i | x_i) can be rewritten as q(z_i |w,μ,σ^2), p(z_i | x_i,μ,σ^2) respectively. The relevant formulas are as follows.

$$q_\phi(z_i|x_i) = N\left(z_i|\mu_i, \sigma_i^2\right), \tag{1}$$

$$z \backsim p_\theta(z|x) = N\left(\mu, \sigma^2\right), \tag{2}$$

$$q_\phi(z;w, \mu, \sigma^2) = \sum_{i=1}^{m} w_i N\left(z_i;\mu_i, \sigma_i^2\right), \tag{3}$$

$$p_\theta(z|x, \theta, \tilde{\mu}, \tilde{\sigma}^2) = \sum_{i=1}^{m} \pi_i N\left(z_i;\tilde{\mu}_i, \tilde{\sigma}_i^2\right), \tag{4}$$

where μ_i, ${\sigma_i}^2$ denote mean and variance respectively, z_i is the hidden embedding feature,w and θ are the the global optimum parameters. The recognition model q(z|x) is the encoder capture z from x. m denotes the number of gaussian distribution. When m=1, it is sample distribution like VAE [12].

3.2 Architecture of BaDEC

As illustrated in Fig. 2, the overall architecture of BaDEC can be divided into two parts, namely encoder and decoder. Figure 2(b) shows our network optimization and Fig. 2(c) shows the schema of ELBO. Algorithm 1 shows a brief description of BaDEC algorithm. The detailed architecture description of the BaDEC framework is as follows.

Encoder

Input layer. The original image or text data is fed to the input layer. Image data is set according to pixel dimensions, and reshape the input dimension based on image sharp.

When deal with text data, we sharp the sequence text to Euclidean matrix space. We set the longest sample as the dimension of the Eculidean matrix column, and the dimension of word vector is the row dimension of Eculidean matrix space. In a word, we construct text data as a shape similar to that of image data.

Embedding layer. We utilize a multi-layer convolutional neural network as the embedding layer. The embedding layer is stacked by convolutional layer and pooling layer. The parameters of layers and activation function are introduced in detail in the experiments.

Calculation layer. This layer calculates the mean and variance of the output features. To avoid overfitting, we constrain the sample by adding noise that follows normal distribution.

Hidden layer. A hidden layer aggregates the output of calculation layer and noise. In the encoder-decoder framework, the hidden layer generates feature representation better for next clustering.

Category prediction layer. We use a deep regularized clustering (Details are shown in Section 3.3) to predict the category from the hidden feature representation. A Softmax layer employs the hidden layer as input to predict category. To optimize the model, we construct a relative entropy constraint term i.e.(KL) to optimize the prediction results iteratively. In addition, in order to adjust the tightness of the boundary, we introduce an adjustment factor λ. We use λ to combine two types of lower evidence boundary optimization methods to achieve the adjustable boundary (Details are shown in Section 3.4).

Decoder:

Embedding layer. The embedding layer of the decoder is symmetrical to that of the encoder. We use the de-convolution structure with the same parameters. The main task of this layer is to restore the input of the model through the features of the hidden layer, which is called the reconstruction stage.

Output layer. As the output of the decoder embedding layer, the sample x' of the output layer reconstructs the sample x of the input layer. The output sample x' is used to form the reconstruction loss with the input sample, which is one of the components of the encoder-decoder framework.

3.3 Deep regularized clustering of BaDEC

In an ingenious view, [5] presents a regularized clustering based on a single Softmax node. Whereas, one Softmax node cannot get stable forecast results from the complex feature space. Thus, we develop a novel regularized clustering in our BaDEC. We utilize n Softmax nodes instead of the single layer obtains more accurate prediction results. We define the clustering layer as follows.

$$p_{ik} = \frac{1}{n} \sum_{n} soft\max(z_i + a_i) \tag{5}$$

$$= \frac{1}{n} \sum_{h=1}^{n} P(y_i = k|(z_i + a_i), \theta_h) \tag{6}$$

$$= \frac{1}{n}\sum_{h=1}^{n}\frac{\exp\left(\theta_k^T\left(z_i + a_i\right)\right)}{\sum_{k'=1}^{k}\exp\left(\theta_{k'}^T\left(z_i + a_i\right)\right)}, \tag{7}$$

where $P(y_i=k|\ z_i)$ denotes the probabilistic of sample i belongs to category k. $\theta=[1,2,..,k]\in R^{dz}$, θ denotes the parameters of Softmax function. For the hidden layer vectors z by encoder, we define clustering prediction layer using a Softmax logistic regression function f_θ: $Z \longrightarrow Y$ to pick the cluster for each sample. Similar to [5], we expect to introduce a regularization term on the target variable and propose the empirical label distribution of the target variable. By predicting the constraints, we can learn high-confidence representations. The f_k and q_{ik} can be defined as (8) and (9).

$$f_k = P(y=k) = \frac{1}{n}\sum_i q_{ik}. \tag{8}$$

$$q_{ik} = \frac{p_{ik}\Big/\left(\sum_{i'} p_{i'k}\right)^{\frac{1}{2}}}{\sum_{k'} p_{ik'}\Big/\left(\sum_{i'} p_{i'k'}\right)^{\frac{1}{2}}}. \tag{9}$$

Besides, we define a relative entropy constraint term $KL(p_{ik} \parallel q_{ik})$, which is to make the predicted categories more balanced. By category prediction layer, we can avoid the similarity calculation between samples, and obtain the difference of samples automatically. Therefore, this design contributes to more precise clustering prediction results generation.

3.4 Boundary-aware variational auto-encoder of BaDEC

Our proposed BaDEC is based on variational auto-encoder. In this subsection, we first define the next generation of variational evidence, so as to set training objective for the model. The evidence lower bound is defined in VAE as (10).

$$\begin{aligned} &ELBO_{VAE}(\theta,\phi,x)\\ &= \int q_\phi(z|x)\log\frac{p_\theta(x,z)}{q_\phi(z|x)}dz\\ &= \log p_\theta(x) - KL(q_\phi(z|x)|p_\theta(x,z)). \end{aligned} \tag{10}$$

To achieve boundary-aware variational auto-encoder of our framework, we then redefine the $ELBO_{GVAE}$ as (11).

$$\begin{aligned} &ELBO_{GVAE}(\theta,\phi,x)\\ &= \int q_\phi\left(z;w,\mu,\sigma^2\right)\log\frac{p_\theta\left(z|x,\theta,\tilde{\mu},\tilde{\sigma}^2\right)}{q_\phi\left(z;w,\mu,\sigma^2\right)}dz\\ &= \log p_\theta\left(z|x,\theta,\tilde{\mu},\tilde{\sigma}^2\right) - KL\left(q_\phi(z;w,\mu,\sigma^2)|p_\theta\left(z|x,\theta,\tilde{\mu},\tilde{\sigma}^2\right)\right)\\ &\le \log p_\theta(x). \end{aligned} \tag{11}$$

$$KL(p(x)||q(x)) = \int p(x) \ln \frac{p(x)}{q(x)} dx \geq 0. \tag{12}$$

Based on the above definitions, our variable bound is designed by a dynamic combination to obtain adjustable evidence lower bounds. In other words, we define the $ELBO_{BaDEC}$ of our BaDEC by considering $ELBO_{VAE}$ and $ELBO_{GVAE}$ with λ, and we set $0 \leq \lambda \leq 1$, as (13).

$$\begin{aligned} & ELBO_{BaDEC}(\theta, \phi, x) \geq \lambda \log p_\theta(x) \\ & + (1-\lambda) \log p_\theta\left(z|x, \theta, \tilde{\mu}, \tilde{\sigma}^2\right) \\ \Rightarrow & \log p_\theta\left(z|x, \theta, \tilde{\mu}, \tilde{\sigma}^2\right) \leq ELBO_{BaDEC}(\theta, \phi, x) \\ \leq & \log p_\theta(x). \end{aligned} \tag{13}$$

3.5 Network optimization

In this subsection, we construct boundary-aware variable evidence lower bounds flexibly, by minimizing KL(q(z_i |w,u,σ^2)‖ p(z_i | x_i,ũ,σ^2)), KL(p_{ik} ‖ q_{ik}), and ‖ $x - x'$‖$^{1\backslash 2}$. The reconstruction loss of the model is defined as (14).

$$loss_{res} = ||x_i' - x_i||^{\frac{1}{2}}. \tag{14}$$

We define the overall objective function as (15).

$$\min \left(\begin{array}{l} \lambda(KL(q_\phi(z|x)|p_\theta(x,z)) \\ +(1-\lambda)KL(q_\phi(z;w,\mu,\sigma^2)||p_\theta(z|x,\theta,\tilde{\mu},\tilde{\sigma}^2)) \\ +KL(q_{ik}||p_{ik}) + loss_{res} \end{array} \right). \tag{15}$$

During the sampling is non-derivable, our training strategy can be seen as a Reparameterization Trick [12] and we use the SGVB to estimate, for z_i=μ+σe, where e is an auxiliary noise variable e~N(0,1). According to the Reparameterization Trick, KL(q(z_i |w,u,σ^2)‖p(z_i | x_i,ũ,σ^2)) can be rewritten as (16).

$$\log q_\phi(z|x^{(i)}) = \log \mathrm{N}\left(z; u^{(i)}, \sigma^{2(i)} I\right). \tag{16}$$

Finally, we use the Reparameterization Trick from [12] to transform the sample distribution into a derivable object, then we use SGVB to optimize the objective, the objective can be rewritten as (17).

$$\begin{aligned} L = \min(\lambda & \left(-\log {\sigma^2}^{(i)} + {\mu^2}^{(i)} + {\sigma^2}^{(i)} - 1 \right) \\ & + (1-\lambda) \frac{1}{2} \sum_{j=1}^{J} \left(-\log {\sigma_j}^{2(i)} + {\mu_j}^{2(i)} + {\sigma_j}^{2(i)} - 1 \right) \\ & + q_{ik} \log \frac{q_{ik}}{p_{ik}} + ||x_i' - x_i||^{\frac{1}{2}}). \end{aligned} \tag{17}$$

Table 1 Statistics of the selected six datasets

Dataset	Examples	Classes	Sample size	Data type
MNIST	60000	10	28×28	image
Fashion-MNIST	60000	10	28×28	image
CIFAR-10	60000	10	28×28×3	image
USPS	9298	10	16×16	image
20NEWS	2965	4	300 d-embedding	text
REUTERS	10000	4	300 d-embedding	text

4 Experiments

In this section, we first introduce the datasets, then describe the details of the experiment, it mainly includes the parameters setting, evaluation metrics, and experimental results and analysis.

Algorithm 1 Deep Boundary-aware Clustering

Input: texts: $T = \{T^1, T^2, ..., T^n\}$;or images: $V = \{V^1, V^2, ..., V^n\}$; Number of clusters k; Number of iterations Iter.
Output: Final clustering results.
1: Load the parameters of text word embedding or image shape;
2: Initialize parameters;
3: **for** $l = 0$ to Iter-1 **do**
4: Generate hidden feature z by Encoder;
5: Calculate the probability of sample i belongs to category k by Eq.(5) from the feature space z;
6: Calculate clustering constraint q_{ik} by Eq.(8) ;
7: Calculate reconstruction loss $loss_{res}$ by Eq.(14);
8: Calculate the overall loss by Eq.(15);
9: Return loss by Eq.(16)
10: Update the parameters of auto-encoder by minimizing Eq.(14) by SVGD optimizer;
11: Get the final results by Eq.(5)

4.1 Datasets

To evaluate the performance of BaDEC, follow these existing excellent deep clustering work [7, 16, 24, 26], we employ six benchmark datasets are to test the performance of clustering. In particular, MNIST, Fashion-MNIST, USPS and CIFAR-10 are image datasets; 20NEWS and REUTERS are text datasets. The statistics of six datasets as shown in Table 1.

- **MNIST**: The MNIST[1] dataset is consists of 10 classes handwritten digits. MNIST is the most popular image clustering, classification datasets, the main characteristics of the sample is clear, simple background.
- **Fashion-MNIST** [23]: A MNIST-like fashion product database,[2] and consists of 10 categories fashion product. Mainly fashion products such as clothes, shoes and bags, and more difficult to distinguish than MNIST.

[1] http://yann.lecun.com/exdb/mnist/
[2] https://github.com/zalandoresearch/fashion-mnist

- **CIFAR-10**: The CIFAR-10 datasets consists of 10 categories, including airplane, automobile, bird, and cat, etc. The intersection between the categories is empty.
- **USPS**: The USPS dataset contains 9298 grayscale images, obtained from the scanning of handwritten digits from envelopes by the U.S. postal service.
- **20NEWS**: 20Newsgroups[3] is a popular database for text classification or clustering. We used four categories: comp.graphics, sci.electronics, talk.politics.guns, rec.motorcycles.
- **REUTERS**: Reuters[4] dataset has 810,000 English news, following DEC, we used 4 categories: corporate, government markets, and economics.

4.2 Experiment setting

In BaDEC, we design the structure of encoder of embedding layers as a multi-layer convolutional neural network for all sets, for the CNN-layer as $Conv_l^k$, l is the number of filter and k is the size of the kernel, the stride sets 3. In encoder embedding, we set $Conv_{64}^3$ -$Conv_{64}^3$-$Conv_{128}^5$-$Conv_{128}^5$-$Conv_{128}^5$, where d is the dimension of input data. The decoder embedding layer is CNN layers same as the encoder. We set dropout is 0.5, mini-batch is 128. In text clustering task, we use BERT [3] pre-training model to initialize word embedding. For different lengths of text, we use tail alignment for text data, and the dimension of word embedding is d_w=300. The learning rate is 0.01, and use SGVB optimizer. The hidden layer space dimension z_i sets 40. The number of distributions m∈[1, 50] in the mixed Gaussian distribution. The n Softmax nodes of regularized clustering layer, n = 8. The embedding layer has 5-layer convolutional neural networks, the optimal bound parameter $\lambda = 0.5$.

4.3 Evaluation metrics

The evaluation of the clustering task is to measure the correctness and purity of the category. Follow these recent works [7, 16, 24, 26], we use clustering accuracy(ACC) and normalized mutual information(NMI) to evaluate the performance of the model. ACC is defined as (17).

$$ACC = \max_m \frac{\sum_{i=1}^n 1\{l_i = m(c_i)\}}{n} \tag{18}$$

NMI is defined as (18).

$$NMI(l, c) = \frac{I(l, c)}{\frac{1}{2}[H(l) + H(c)]} \tag{19}$$

4.4 Baseline methods

To evaluate the performance of BaDEC, we compare it with several popular baseline method. In particular, to verify the effectiveness of our representation learning, we set up

[3] http://qwone.com/jason/20Newsgroups/

[4] https://github.com/philipperemy/Reuters-full-data-set

four comparison methods based on K-Means. For the accuracy of the overall clustering task, we select five comparison models. The detailed description of comparison baseline methods are as follows.

- **K-Means**: As a classic clustering method, it is widely used to compare the performance of clustering. We run K-Means [27] algorithm in the original feature space.
- **K-Means+AE**: In order to reflect the effectiveness of feature-Autoencoder, we employ the K-Means algorithm to pick cluster from feature that generated by auto-encoder.
- **K-Means+VAE**: In order to reflect the effectiveness of feature-Variational AutoEncoder, we run K-Means algorithm in the VAE [12] feature space.
- **K-Means+BaDEC**: In order to show the advantages of our features, we run K-Means algorithm in the BaDAC(ours) hidden embedding space.
- **DEC** [24]: The deep embedded clustering learns feature representations and cluster assignments by using deep neural networks. We use the authors' released code of DEC, and the parameter settings follow the original settings.
- **IDEC** [7]: IDEC is an improved version of DEC with local structure preservation. We set the parameters and hyper-parameter the same as DEC.
- **SDEC** [19]:It is a semi-supervised deep embedding clustering that jointly optimize cluster labels assignment and learn features.
- **VaDE** [11]: VaDE is an unsupervised and generative approach to clustering that combines gaussian mixture model and variational auto-encoder.
- **ClusterGAN** [16]: ClusterGAN is an architecture that enables clustering in the latent space, and the latent space is created by a mixture of discrete and continuous latent variables.
- **BaDEC** ($\lambda = 1$): When the parameter $\lambda = 1$, our BaDEC is to optimize a standard evidence lower bound similar to DEC.
- **BaDEC** ($\lambda = 0$): When the parameter $\lambda = 0$, our BaDEC is to optimize a tighter evidence lower bound similar to IDEC.
- **BaDEC** ($0 < \lambda < 1$): When the parameter $0 < \lambda < 1$, our BaDEC adjusts the tightness of the bound by adjusting the value of lambda. The experimental result is optimal obtained by adjusting the parameters λ.

4.5 Experimental results and analysis

We evaluate the performance of the model from the aspect of feature learning and overall performance of clustering. We report the result of baseline methods and BaDEC in Table 2 with ACC and NMI and we show partial clustering visualization results for Fashion-MNIST and MNIST dataset in Fig. 3. We analyze the experimental results from three aspects as follows. In order to compare the performance of the model from multiple aspects, we argue three questions.

RQ1. Can our model efficiently complete feature representation learning? If yes, do we learn better features than other methods?

RQ2. Whether our method can achieve excellent clustering performance? If yes, what are the advantages that we have over existing methods?

Table 2 The results of our proposed framework and several baseline methods on six benchmark datasets

Methods	MNIST(%)		Fashion(%)		CIFAR-10(%)		USPS(%)		20NEWS(%)		REU-TERS(%)	
Metrics	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
K-Means	55.4	53.8	47.5	51.5	21.8	10.1	65.3	62.8	33.76	0.71	53.3	52.7
K-Means+AE	78.6	71.2	48.2	51.3	27.4	23.9	68.2	61.4	40.8	18.6	71.3	64.9
K-Means+VAE	79.3	71.6	50.3	51.4	28.0	23.4	70.3	61.1	45.3	23.1	73.5	70.7
K-Means+BaDEC(z_i)	83.5	78.1	62.3	56.4	30.6	25.3	71.1	62.3	49.6	27.9	76.3	69.2
DEC [24]	84.3	77.6	51.6	54.6	26.3	25.7	74.1	74.3	50.1	44.4	75.6	70.4
IDEC [7]	88.4	86.7	52.9	55.7	25.1	24.7	76.2	78.5	53.6	44.5	77.4	69.2
ClusterGAN [16]	95.0	**89.0**	63.0	64.0	45.2	**40.1**	84.9	82.4	78.7	**77.3**	83.1	79.5
SDEC [19]	86.1	82.9	54.5	51.6	27.3	17.2	76.4	77.7	78.12	46.4	80.5	78.4
VaDE [11]	94.5	87.6	55.2	57.3	36.8	34.1	56.6	51.2	67.4	43.5	80.9	77.3
BaDEC ($\lambda = 1$)	84.6	78.5	53.6	50.5	34.6	31.7	75.2	73.6	61.7	56.7	79.1	74.7
BaDEC ($\lambda = 0$)	89.1	86.5	56.3	54.9	38.5	37.7	79.6	77.2	65.3	60.6	81.5	76.4
BaDEC ($\lambda = 0.5$)	**96.3**	88.6	**63.7**	**67.2**	**49.7**	34.6	**87.4**	**85.8**	**79.7**	74.5	**83.6**	**81.3**

The best results are marked with bold symbol

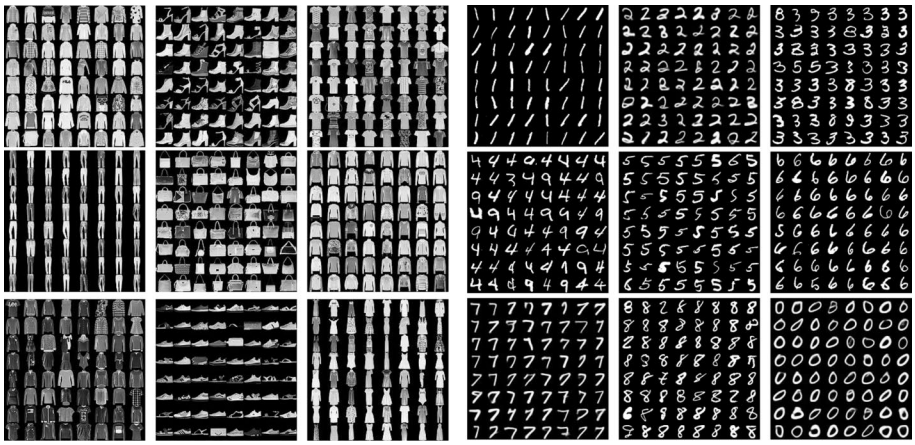

Fig. 3 The partial clustering visualization results on Fashion-MNIST and MNIST datasets

RQ3. Is the parameter λ we set valid? If yes, what impact will the value of the parameter have on the performance of the model?

Experiment 1 To answer RQ 1. Effect of representation learning To estimate the representation performance of our model, we set up four comparison methods based on K-Means. We use K-Means algorithm to cluster calculation by different feature representation spaces. It can be seen from the experimental results as shown in Table 2, despite we set the same feature dimension, the complex embedding space of the model is more suitable for clustering tasks. Compared with the original feature space, the clustering results of AE, VAE, and BaDEC have great improvement. This is due to the fact that encoding framework can learn more effective representation of the features of the

sample. Besides, the original sample contains too much redundant information, which may cause inaccurate results in the process of similarity calculation. In contrast, the VAE's embedding space is much better than the original space, but there are some points between the categories that are not easy to resolve, which can cause computational errors.

Table 2 shows that our BaDEC has great improvement compared to AE and VAE in feature representation space capabilities. Several observations can be made. In encoder-decoder frameworks without adding adversarial sham are easy fall into overfitting. First, both VAE and BaDEC are generative models, because the noise term is added, which can make the two stages of encoding and decoding adversarial learning. Due to the noise, the model avoids the overfitting. By comparing models without generative capabilities, VAE and BaDEC have learned richer feature representation. Therefore, VAE and BaDEC have certain advantages in feature learning than AE. Compared to VAE, our BaDEC constructs an unified framework that performs both feature learning and cluster prediction tasks simultaneously. By a joint training object, BaDEC gets the high-confidently feature representation. Besides, BaDEC uses a variable optimization bound to better optimize the evidence boundary. In the cluster loss term, the model constructs a clustering constraint and the iterative optimization feature space via iteration learning. In a word, the BaDEC can learn clustering high-confidently feature representation through the constructed framework, and complete the cluster prediction task simultaneously.

Experiment 2 To answer RQ 2. Overall performance of clustering For the overall clustering effect of the model, we select five popular clustering models, DEC, IDEC, SDEC, ClusterGAN and VaDE. DEC is a representative method which can perform feature space learning and cluster analysis on real data simultaneously. IDEC and SDEC are optimized models based on DEC, and the clustering accuracy is improved by 4.1% and 2.9% on maximum compared to DEC model respectively. Compared with models based on DEC, VaDE utilizes the Gaussian mixture distribution as clustering prediction, then iteratively performs clustering feedback the process of embedding feature leaning. VaDE avoids to utilizes partition clustering method, and it uses a mixed Gaussian distribution to constrain the sample. Compared with VAE-based and VaDE models, BaDEC propose a prediction mechanism to replace the clustering algorithm which mitigates the unreliability of similarity calculation. As observed in Table 2, compared to SDEC, the average improvements achieved by BaDEC are 10.9% for ACC, for 14.9% NMI. ClusterGAN is a generative model based on generative adversarial networks, BaDEC archives broadly better results than it. It is because ClusteringGAN is a kind of adversarial work, our BaDEC is weaker on the two NMI values, MNIST and 20NEWS. In this view, BaDEC obtains the differences between samples through the learning of neural networks. To mitigate the complexity and unreliability of similarity calculations, BaDEC employs a joint optimization object to train the framework. Thus, BaDEC has greatly improvement on the overall task of clustering compare to the baseline methods. Compare to BaDEC ($\lambda = 1$) and BaDEC ($\lambda = 0$), BaDEC by adjusting the value of λ, better performance is obtained. Once again proved the advantage of boundary-aware.

Experiment 3 To answer RQ3. Parameter sensitivity analysis Compared with the VaDE model, in the BaDEC framework, similar to VaDE, we also use a mixed distribution in the framework. We sample from mixed Gaussian distribution and add the λ to change the tightness of the evidence bound which can better perform feature embedding learning. Also, we use clustering constraints to constrain the model training, making the clustering results more accurate. Figures 4 and 5 have shown the impact of the parameter λ on the REUTERS and USPS datasets. It shows that the variation of ACC and NMI with epoch sizes. As the

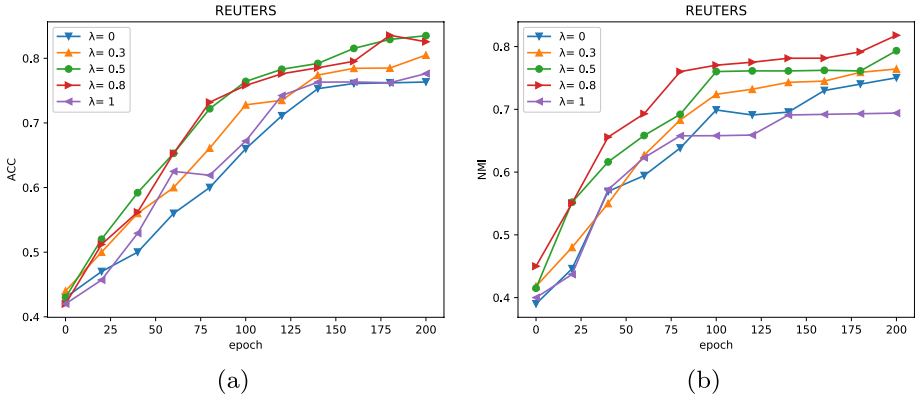

Fig. 4 Impact of the parameter λ on REUTERS dataset. It shows that the variation of ACC and NMI with epoch sizes

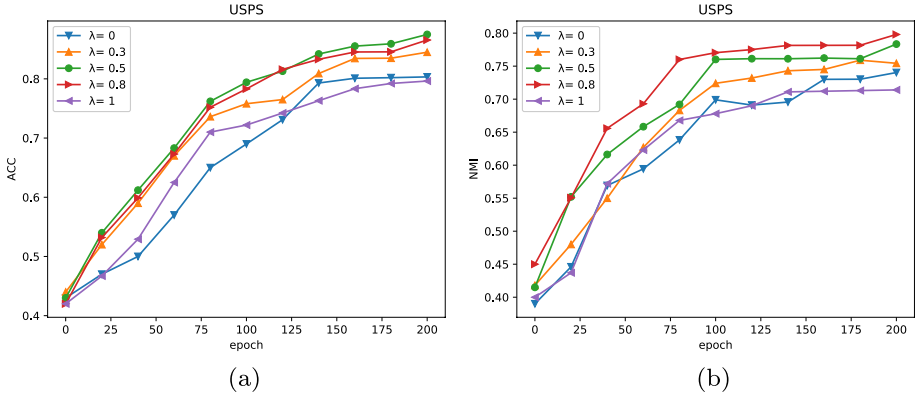

Fig. 5 Impact of the parameter λ on USPS dataset. It shows that the variation of ACC **a**,**c** and NMI **b**,**d** with epoch sizes

results we find that when λ is between 0.3 and 0.8, the optimization is better. We analyze BaDEC in terms of the ability to feature embedding space and the overall clustering performance. The results show that our model has greatly improvement than the baseline models. Thus, the feature space is good enough, then the classification layer of a prediction layer can accurately predict the category of the sample. The experimental results verify that the effectiveness of our BaDEC.

5 Conclusion and further work

In this paper, we aim to design a deep clustering framework based on boundary-aware variational auto-encoder and automated clustering. In the framework of BaDEC, we can learn feature embedding and clustering tasks simultaneously. To avoid direct similarity calculation between samples, we employ automated clustering mechanism to measure

the difference between samples. Especially, we design a boundary-aware variational autoencoder to adjust evidence lower bounds automatically for deep clustering. Finally, we define a constructing combination objective optimization function to train our model better. The experimental results verify that our proposed BaDEC achieves significantly improvement on various datasets compared to popular baseline methods. In other words, the method we propose is a general framework. However, there are certain limitations. These limitations are mainly the common challenges of deep clustering methods. For example, data with better data distribution tends to get better results. However, the complexity of data distribution is still a very serious challenge. In the future, we will analyze more complicated data (for example, graph data of social network) to gain accurate representation for deep clustering.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Grant No. 61602353 and 61373148), Hubei Provincial Natural Science Foundation of China (Grant No. 2017CFA012), the National Social Science Foundation under Award (Grant No. 19BYY076), in part Key R & D project of Shandong Province (Grant No. 2019JZZY010129), and Shandong Provincial Social Science Planning Project (Grant No. 18CXWJ01, 18BJYJ04 and 19BJCJ51).

References

1. Boutsidis C, Drineas P, Mahoney MW (2009) Unsupervised feature selection for the *k*-means clustering problem. In: Advances in neural information processing systems, pp. 153–161
2. Chen X, Kingma DP, Salimans T, Duan Y, Dhariwal P, Schulman J, Sutskever I, Abbeel P (2017) Variational lossy autoencoder. In: 5th international conference on learning representations, Toulon, France, April 24-26
3. Devlin J, Chang MW, Lee K, Toutanova K Bert (2019) Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: human language technologies, Minneapolis, MN, USA, June 2-7, pp. 4171–4186
4. Fu Y, Yan Q, Liao J, Xiao C (2020) Joint texture and geometry optimization for RGB-D reconstruction. In: 2020 IEEE/CVF conference on computer vision and pattern recognition, Seattle, WA, USA, June 13-19, pp. 5949–5958
5. Ghasedi Dizaji K, Herandi A, Deng C, Cai W, Huang H (2017) Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In: Proceedings of the IEEE international conference on computer vision, pp. 5736–5745
6. Gregor K, Besse F, Rezende DJ, Danihelka I, Wierstra, D (2016) Towards conceptual compression. In: Advances in neural information processing systems 29: Annual conference on neural information processing systems, Barcelona, Spain, December 5-10, pp. 3549–3557
7. Guo X, Gao L, Liu X, Yin J (2017) Improved deep embedded clustering with local structure preservation. In: IJCAI, pp. 1753–1759
8. Huang C, Sankaran K, Dhekane E, Lacoste A, Courville AC (2019) Hierarchical importance weighted autoencoders. In: Proceedings of the 36th international conference on machine learning, Long Beach, California, USA, June 9-15, pp. 2869–2878
9. Ji P, Zhang T, Li H, Salzmann M, Reid, I (2017) Deep subspace clustering networks. In: Advances in neural information processing systems, pp. 24–33
10. Jiang B, Tu W, Yang C, Yuan J (2020) Context-integrated and feature-refined network for lightweight object parsing. IEEE Trans Image Process 29:5079–5093
11. Jiang Z, Zheng Y, Tan H, Tang B, Zhou H (2017) Variational deep embedding: An unsupervised and generative approach to clustering. In: Proceedings of the twenty-sixth international joint conference on artificial intelligence, pp. 1965–1972
12. Kingma DP, Welling M (2014) Auto-encoding variational bayes. In: 2nd international conference on learning representations
13. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105

14. Liu D, Long C, Zhang H, Yu H, Dong X, Xiao C (2020) Arshadowgan: Shadow generative adversarial network for augmented reality in single light scenes. In: 2020 IEEE/CVF conference on computer vision and pattern recognition, Seattle, WA, USA, June 13-19, pp. 8136–8145
15. Martínez-Trinidad JF, Ruiz-Shulcloper J (2001) Fuzzy clustering of semantic spaces. Pattern Recognition 34(4):783–793
16. Mukherjee S, Asnani H, Lin E, Kannan S (2019) Clustergan: Latent space clustering in generative adversarial networks. In: Proceedings of the thirty-third AAAI conference on artificial intelligence, pp. 1181–1190
17. Ng AY, Jordan MI, Weiss Y (2002) On spectral clustering: Analysis and an algorithm. In: Advances in neural information processing systems, pp. 849–856
18. Rainforth T, Kosiorek AR, Le TA, Maddison CJ, Igl M, Wood F, Teh YW (2018) Tighter variational bounds are not necessarily better. In: Proceedings of the 35th international conference on machine learning, pp. 4274–4282
19. Ren Y, Hu K, Dai X, Pan L, Hoi SC, Xu Z (2019) Semi-supervised deep embedded clustering. Neurocomputing 325:121–130
20. Solorio-Fernández S, Martínez-Trinidad JF, Carrasco-Ochoa JA (2017) A new unsupervised spectral feature selection method for mixed data: a filter approach. Pattern Recognition 72:314–326
21. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research 11:3371–3408
22. Xiang S, Nie F, Zhang C (2008) Learning a mahalanobis distance metric for data clustering and classification. Pattern recognition 41(12):3600–3612
23. Xiao H, Rasul K, Vollgraf R (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747
24. Xie J, Girshick R, Farhadi A (2016) Unsupervised deep embedding for clustering analysis. In: International conference on machine learning, pp. 478–487
25. Xing EP, Jordan MI, Russell, SJ, Ng AY (2003) Distance metric learning with application to clustering with side-information. In: Advances in neural information processing systems, pp. 521–528
26. Yang J, Parikh D, Batra D (2016) Joint unsupervised learning of deep representations and image clusters. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5147–5156
27. Ye J, Zhao Z, Wu M (2008) Discriminative k-means for clustering. In: Advances in neural information processing systems, pp. 1649–1656
28. Yu J, Jiang H, Wang G, Guo Q (2012) Clustering-based energy-efficient broadcast tree in wireless networks. Int. J. Comput. Commun. Control 7(4):785–790
29. Zhang R, Tong H, Xia Y, Zhu Y (2019) Robust embedded deep k-means clustering. In: Proceedings of the 28th international conference on information and knowledge management, Beijing, China, November 3-7, pp. 1181–1190
30. Zhou F, De la Torre F, Hodgins JK (2012) Hierarchical aligned cluster analysis for temporal clustering of human motion. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(3):582–596