IDECF: IMPROVED DEEP EMBEDDING CLUSTERING WITH DEEP FUZZY SUPERVISION

Mohammadreza Sadeghi and Narges Armanfard

Department of Electrical & Computer Engineering, McGill University, Montréal, Québec, Canada
Mila - Québec AI Institute, Montreal, Québec, Canada
{mohammadreza.sadeghi, narges.armanfard}@mcgill.ca

ABSTRACT

Deep clustering algorithms utilize a deep neural network to map data points in a lower-dimensional space which is more suitable for clustering task. Recent algorithms employ autoencoder to jointly learn a lower-dimensional space (aka latent space) and perform data clustering through minimizing a clustering loss. These algorithms suffer from the fact that the true cluster assignments are unknown because of the unsupervised nature of the task. Thus, they adopt a self-training strategy and estimate the true cluster labels using the algorithm parameters; while the true parameters' value is unknown at the problem outset. To address this difficulty, we propose a deep clustering technique, called IDECF, whereby the true cluster assignments are estimated using an individual deep fully connected network (FCM-Net) which takes its input from the latent space of an autoencoder. The proposed IDECF is trained in an end-to-end manner by minimizing a linear combination of reconstruction loss and clustering loss. Experimental results on benchmark datasets demonstrate the viability and effectiveness of the proposed algorithm.

Index Terms— deep clustering, deep embedding clustering, fuzzy supervision.

1. INTRODUCTION

Clustering methods aim to partition data points based on a similarity metric. Out of the different methods available for clustering, k-means [1] and fuzzy c-means [2] are the two well-known conventional methods that could be applied to a variety of tasks [3, 4, 5] due to their simplicity. However, these conventional methods could only extract local relationship between data points in the original input space and fail to describe latent dependencies between data points. Moreover, because of the nature of the employed distance metrics, these algorithms do not show promising performance when data points are not evenly scattered near cluster centers. Recently, deep-learning-based approaches have been widely studied and employed in various types of applications such as image segmentation [6], social network analysis [7], face recognition [8], and computer vision [9]. Deep clustering approaches endeavor to combine embedding and clustering to boost clustering performance of conventional methods. Most deep clustering methods make use of autoencoders (AEs) to capture nonlinear dependencies between data points. The encoder part of an AE transforms (aka embeds) data points in a lower-dimensional space (aka latent space) and the decoder part aims to reconstruct original data points using the latent space. Encoder and decoder can be trained by minimizing a reconstruction loss, which measures the degree of similarity between original input and reconstructed output. In a data clustering problem, the true cluster assignment of data points is unknown. Some algorithms such as deep embedding clustering (DEC) [10] employ a self-training procedure [11] to estimate the true cluster assignments; the estimations are then used as navigators when training algorithm's parameters. In DEC, an autoencoder is first pre-trained; then, after discarding the AE's decoder part, the algorithm focuses on fine-tuning the encoder's parameters through minimizing clustering loss, where the clustering loss is highly affected by the estimations obtained for the true cluster assignments. Some other algorithms, such as deep clustering network (DCN) [12]

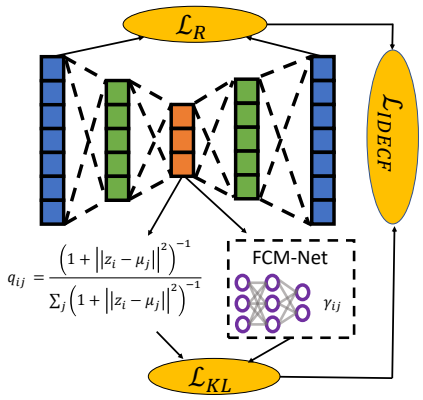

Fig. 1. Overall training procedure of the proposed IDECF method.

The authors acknowledge the financial contributions of MITACS Canada. The assistance of Ericsson, Montreal, Canada is also gratefully acknowledged.

and deep k-means (DKM) [13], improve clustering performance by integrating reconstruction loss with clustering loss when training algorithm parameters; including both of these losses help the algorithm to maintain local structure and relationship between original data points and improve clustering performance.

In this research study, we propose an improved deep embedding clustering technique with deep fuzzy supervision (IDECF). The proposed IDECF improves the DEC algorithm [10] through two steps: 1) despite DEC that estimates cluster assignments through a self-training strategy, IDECF estimates cluster assignments (aka target distribution) using a proposed deep fuzzy c-means network (FCM-Net) which is specifically designed and trained for this purpose. 2) Inspired by [12, 13], IDECF makes use of the AE's decoder part and incorporates reconstruction loss besides clustering loss; this contributes to local structure preservation of the data points. Fig 1. shows the overall training scheme of the proposed IDECF algorithm.

Details of the IDECF method is presented in Section 2. Performance of the IDECF on benchmark datasets is shown in Section 3.

2. PROPOSED METHOD

In this research study, we aim to address the problem of grouping dataset $X = \{x_1, x_2, ..., x_N\}$, with N samples, into K disjoint clusters. The proposed IDECF consists of an autoencoder and a FCM-Net. The autoencoder comprises of encoder and decoder networks respectively denoted by $f(.)$ and $g(.)$. Latent representation of X in the embedding space is shown by $Z = \{z_1, z_2, ...z_N\}$, where $z_i = f(x_i; \theta_e) \in R^d$ for $i = 1, ..., N$, θ_e and d respectively denotes the encoder parameters and the latent space dimension. The autoencoder output for input data x_i is represented by $\hat{x}_i = g(z_i, \theta_d)$, where θ_d is parameters of the decoder network. The jth cluster center is denoted by μ_j, $j = 1, \ldots, K$. FCM-Net's parameters set is denoted by θ_{FCM}.

In the remaining, we first briefly present the DEC algorithm in Section 2.1; the proposed IDECF structure is then presented in Sections 2.2 and 2.3.

2.1. Deep embedding clustering

Deep embedding clustering (DEC) [10] begins with pretraining an autoencoder; the initial values for the jth cluster centre μ_j is obtained by applying k-means to the latent representation of data in the AE's embedding space. It then discards the decoder part and fine-tunes the encoder part to get a more effective lower-dimensional space for the data clustering task. The encoder part endeavors to minimize Kullback–Leibler (KL) divergence loss function defined as below:

$$\mathcal{L}_{DEC} = KL(P||Q) = \sum_i \sum_j p_{ij} \log(\frac{p_{ij}}{q_{ij}}), \qquad (1)$$

where soft assignment, q_{ij}, measures similarity between latent representation of the ith data point z_i and the jth cluster center μ_j using Student's t-distribution as is shown in (2a). p_{ij} is target distribution and is defined in (2b) where it can be seen that the DEC method employs soft assignments to define target distributions; i.e., it uses a self-training approach [11] when minimizing $\mathcal{L}_{DEC}$. Finally, DEC updates cluster centers to minimize $\mathcal{L}_{DEC}$ using stochastic gradient descent (SGD) as is shown in (2c).

$$q_{ij} = \frac{(1+||z_i - \mu_j||^2)^{-1}}{\sum_j (1+||z_i - \mu_j||^2)^{-1}} \qquad (2a)$$

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j (q_{ij}^2 / \sum_i q_{ij})} \qquad (2b)$$

$$\mu_j = \mu_j - \alpha \frac{\partial \mathcal{L}_{DEC}}{\partial \mu_j} \qquad (2c)$$

2.2. Target distribution estimation using FCM-Net

Despite DEC that uses soft assignments for target distribution estimation, we propose to dedicate a fully connected neural network (FCM-Net) for target distribution estimation. FCM-Net is trained independent from the soft assignments, in an iterative manner over the training data.

The proposed FCM-Net aims to estimate true cluster assignments (i.e. target distribution) for different data points. The input space for this network is the latent space of IDECF's autoencoder, i.e. Z. The output of FCM-Net has K neurons, where each neuron corresponds to one cluster; we employ a softmax function at the output layer to have a probability value for assignment to each cluster. In other words, the jth output neuron of the FCM-Net, denoted by γ_{ij}, indicates the target distribution value for the ith data point and the jth cluster. FCM-Net is trained to minimize the well-known fuzzy c-means objective function, shown in (3), for each data sample x_i; where $\mathfrak{B}$ is a batch of data and m is the level of fuzziness. m is set to 1.5 in all our experiments.

$$\mathcal{L}_{FCM} = \sum_{x_i \in \mathfrak{B}} \min_{\gamma_{ij}} \sum_{j=1}^{K} \gamma_{ij}^m ||z_i - \mu_j||^2 \qquad (3)$$
$$s.t \quad \sum_{j=1}^{K} \gamma_{ij} = 1$$

Since the FCM-Net network has a softmax function at the last layer, the constraint term shown in (3) is already satisfied; thus, we only need to minimize $\mathcal{L}_{FCM}$ to estimate cluster assignments (aka target distribution values) γ_{ij}.

Note that although a local minimum of (3) can be obtained thorough a closed-form solution (see [2]), however, we proposed to find the solution using the deep fully connected network FCM-Net. This allows us to take into consideration all previously seen data points in addition to the current batch of data $\mathfrak{B}$, when computing target distribution values of the current data batch. Further, it helps us to capture and model the complex relationship between data points when finding target distributions. Hence, FCM-Net is expected to find a more

generalized solution compare to the traditional closed-form one. This improved performance is demonstrated in Section 3.1.

2.3. IDECF algorithm

As is discussed before, IDECF networks consist of an AE and FCM-Net. To initialize the AE parameters, i.e. θ_e and θ_d, we train a vanilla autoencoder, in an end-to-end manner, that aims to minimize reconstruction loss which is defined as the mean squared error between input and output of the AE. We then apply k-means algorithm to the embedded space of the trained AE to initialize cluster centers μ_j, $j = 1, ..., K$, to the centers defined by k-means. FCM-Net's parameters, θ_{FCM}, are initialized based on Xavier uniform initialization technique [14].

To train IDECF networks using data batch $\mathfrak{B}$, we first embed $\mathfrak{B}$ into the latent space of the AE. The embedded values are then used to update θ_{FCM} through minimizing $\mathcal{L}_{FCM}$, over T successive iterations. The trained FCM-Net is then used to calculate the target distribution values γ_{ij} for the data points in batch $\mathfrak{B}$. We use Student's t-distribution to obtain cluster soft assignments, q_{ij}, as is shown in (2a). Inspired by [12, 13], we define total loss function $\mathcal{L}_{IDECF}$ as linear combination of clustering loss and reconstruction loss. We choose to use KL divergence between q_{ij} and γ_{ij} as an estimation of the clustering loss. These losses are shown below in (4), where λ is a hyperparameters that indicates the importance of the KL divergence loss in the optimization process. λ is set to 0.1 in all our experiments. We update the parameters of the AE through minimizing $\mathcal{L}_{IDECF}$. Minimizing the reconstruction loss helps in preserving the local structure of input data points, and minimizing the KL divergence loss endeavors to refine clusters over iterations and therefore enhances clustering performance.

$$\mathcal{L}_{IDECF} = \mathcal{L}_R + \lambda \mathcal{L}_{KL} \tag{4a}$$

$$\mathcal{L}_R = \sum_{x_i \in \mathfrak{B}} \frac{1}{|\mathfrak{B}|} ||x_i - \hat{x}_i||^2 \tag{4b}$$

$$\mathcal{L}_{KL} = \sum_{x_i \in \mathfrak{B}} \sum_j \gamma_{ij} \log(\frac{\gamma_{ij}}{q_{ij}}) \tag{4c}$$

Finally, we update cluster centers μ_j to minimize $\mathcal{L}_{FCM}$ defined in (3), as is shown bellow:

$$\frac{\partial \mathcal{L}_{FCM}}{\partial \mu_j} = 0 \rightarrow \mu_j = \frac{\sum_{x_i \in \mathfrak{B}} \gamma_{ij}^m z_i}{\sum_{x_i \in \mathfrak{B}} \gamma_{ij}^m}. \tag{5}$$

2.4. Final crisp cluster assignment

After completing the training phase of the IDECF networks, we use the trained encoder part and the trained FCM-Net to assign a data point to a cluster (aka crisp assignment). Specifically, for each data point x_i, we first find the latent representation z_i, using the trained encoder part of the IDECF's autoencoder. Then we compute the corresponding target distribution values $\gamma_{ij}, j = 1, \ldots, K$ using the trained FCM-Net. Finally, x_i is assigned to the more probable cluster, i.e. the cluster with the highest target distribution value.

Pseudo code of the proposed IDECF algorithm is presented in Algorithm 1.

Algorithm 1 Pseudo code of the proposed IDECF algorithm.

Input: X, K, $MaxIter$, T, λ, m
Output: θ_e, θ_d, θ_{FCM}, μ_j $j = 1, \ldots, K$
Initialize θ_e, θ_d, θ_{FCM}, μ_j $j = 1, \ldots, K$ (See Section 2)
for $iter \in \{1, 2, ..., MaxIter\}$ **do**
  Update FCM-Net based on Section 2.2
  Update θ_e, θ_d by minimizing (4a) (See Section 2.3)
  Update cluster centers using (5)
end for
Compute crisp cluster assignments (see Section 2.4)

3. EXPERIMENTS

In this section, the clustering performance of the proposed IDECF method is evaluated on four benchmark datasets. Since the clustering task is an unsupervised task, we concatenate train and test splits of datasets. The datasets are: MNIST [15] which comprises of 60,000 training and 10,000 test samples of 28×28 handwritten gray-scale images; Fashion MNIST [16] that is similar to MNIST in terms of the number of samples and image size, however, it contains images of various types of fashion products, which makes this dataset more complicated for a data clustering task; 2MNIST is a more challenging dataset which is created by concatenation of MNIST and fashion MNIST; hence, it has 140,000 images of 28×28 pixels in 20 classes; USPS [17] contains 9,298 16×16 handwritten images of USPS postal services.

In all our experiments, following DEC, we set IDECF's autoencoder structure to D-500-500-2000-10 for all datasets, where D is the input space dimension. We use the ReLU activation function for all layers except the last and the middle layers that we do not consider any nonlinear function for them. Furthermore, we choose a fully connected neural network for FCM-Net with d-100-200-K neurons for all datasets, where d is set to 10. We use the ReLU activation function for all layers except the last layer where we use the softmax function.

The effectiveness of the proposed IDECF algorithm is compared against conventional and deep-learning-based state-of-the-art clustering algorithms. The conventional algorithms include k-means [1], large-scale spectral clustering (LSSC) [18], and locality preserving non-negative matrix factorization (LPMF) [19]. Our comparison Deep-learning-based methods are DEC [10], DCN [12], and DKM [13]. Furthermore, we demonstrate IDECF performance on AE + kmeans method where k-means algorithm is applied to the latent representation of a vanilla autoencoder (with the same

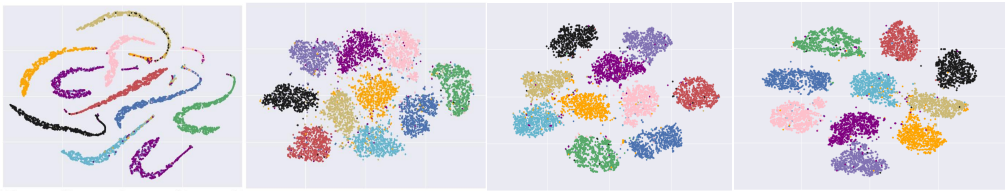

Fig. 2. Visualization of different methods using t-SNE. Axes range from -100 to 100.

architecture as of the IDECF's AE) trained using reconstruction loss. To demonstrate the effectiveness of incorporating $\mathcal{L}_R$ in $\mathcal{L}_{IDECF}$, the performance of the proposed method, when $\mathcal{L}_R$ in (4a) is set to zero, is also reported under the name of IDECF w/o RL. In addition, to show the effectiveness of FCM-Net compare to using the closed-form solution for solving (3), we use the traditional closed-form solution obtained in [2] as the target distribution values γ_{ij} and report the results under the name of IDECF-closed. Performance of the full proposed algorithm (including FCM-Net and $\mathcal{L}_R$) is shown under IDECF.

We use the well-known clustering accuracy metric (ACC) [20] to evaluate and compare the clustering performance of different methods. ACC finds the best match between true cluster labels and predicted crisp cluster assignments by utilizing Hungarian algorithm [21]. The formula for ACC is presented in (6), where l_i and c_i are true and predicted labels for data point x_i, respectively. $\mathbb{1}\{.\}$ is the indicator function.

$$ACC = \max_{m} \frac{\sum_{i=1}^{N} \mathbb{1}\{l_i = m(c_i)\}}{N} \tag{6}$$

3.1. Clustering performance

Table 1 shows the clustering performance of our comparison methods and the proposed IDECF method. we report the performance for comparison methods using the code released by authors of their corresponding research studies with the same hyperparameters mentioned in their works. For each dataset, the best performance is shown in bold. Among all the ten clustering methods, the proposed IDECF achieved the highest performance in all experiments. IDECF method enhanced clustering performance of the baseline methods AE + kmeans and DEC by respectively 5.62% and 3.46%, on average; this verifies the effectiveness of the proposed FCM-Net in learning target distributions over the DEC method that employs a self-training strategy. Moreover, IDECF improved performance of IDECF w/o RL and IDECF-closed by respectively 1.37% and 2.11%; this demonstrates the effectiveness of combining reconstruction loss with KL divergence loss and the effectiveness of using the deep fully connected network FCM-Net for estimating cluster assignments rather than the traditional closed form solution of fuzzy c-means algorithm.

Table 1. ACC on the benchmark datasets for different clustering methods.

Method \ Datasets	MNIST	Fashion MNIST	2MNIST	USPS
k-means	53.20	47.40	32.31	66.80
LSSC	71.40	49.60	39.77	74.60
LPMF	47.10	43.40	34.68	65.20
AE + kmeans	81.26	50.11	39.47	71.90
DEC	84.30	51.80	41.20	74.08
DCN	83.00	51.22	41.35	73.00
DKM	84.00	51.22	41.75	75.40
IDECF w/o RL	86.81	55.61	43.10	74.21
IDECF-closed	87.15	55.93	41.88	71.81
IDECF	**87.17**	**58.63**	**43.68**	**75.73**

3.2. t-SNE visualization

In this section, we show the improved performance of the proposed IDECF algorithm in separating data clusters through visual assessments. Towards this, t-SNE [22] method is applied to the latent representation of each algorithm for the MNIST-test dataset. As it can be seen in Fig 2, IDECF results in a more clear distribution structure over state-of-the-art deep-learning-based approaches. By looking at DEC and the proposed IDECF (Fig 1(a) and Fig 1(d)), one can conclude that training a separated network for learning target distributions leads to higher inter-cluster and lower intra-cluster distances. Furthermore, the advantage of IDECF method over DCN and DKM is more tangible when looking at the separation of clusters in color cyan, olive, and orange that are respectively correspond to digits 3, 5, and 8.

4. CONCLUSION

In this paper, we proposed a practical and effective deep-learning-based method IDECF that endeavors to simultaneously find a low-dimensional representation of data points and perform clustering task. Despite traditional deep-learning-based algorithms that apply self-training to estimate target distribution, in IDECF, in order to improve clustering performance, we propose to train an individual fully connected network FCM-Net. Experimental results show the effectiveness of the proposed IDECF method over state-of-the-art techniques.

References

[1] Stuart Lloyd. “Least squares quantization in PCM”. In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

[2] James C Bezdek, Robert Ehrlich, and William Full. “FCM: The fuzzy c-means clustering algorithm”. In: *Computers & Geosciences* 10.2-3 (1984), pp. 191–203.

[3] OJ Oyelade, OO Oladipupo, and IC Obagbuwa. “Application of k Means Clustering algorithm for prediction of Students Academic Performance”. In: *arXiv preprint arXiv:1002.2425* (2010).

[4] Meng Jianliang, Shang Haikun, and Bian Ling. “The application on intrusion detection based on k-means cluster algorithm”. In: *2009 International Forum on Information Technology and Applications*. Vol. 1. IEEE. 2009, pp. 150–152.

[5] Mahnaz EtehadTavakol, Saeed Sadri, and EYK Ng. “Application of K-and fuzzy c-means for color segmentation of thermal infrared breast images”. In: *Journal of medical systems* 34.1 (2010), pp. 35–42.

[6] John R Hershey et al. “Deep clustering: Discriminative embeddings for segmentation and separation”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 31–35.

[7] Xia Hu, Qiaoyu Tan, and Ninghao Liu. “Deep representation learning for social network analysis”. In: *Frontiers in Big Data* 2 (2019), p. 2.

[8] Mei Wang and Weihong Deng. “Deep face recognition with clustering based domain adaptation”. In: *Neurocomputing* (2020).

[9] Mathilde Caron et al. “Deep clustering for unsupervised learning of visual features”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 132–149.

[10] Junyuan Xie, Ross Girshick, and Ali Farhadi. “Unsupervised deep embedding for clustering analysis”. In: *International conference on machine learning*. 2016, pp. 478–487.

[11] Kamal Nigam and Rayid Ghani. “Analyzing the effectiveness and applicability of co-training”. In: *Proceedings of the ninth international conference on Information and knowledge management*. 2000, pp. 86–93.

[12] Bo Yang et al. “Towards k-means-friendly spaces: Simultaneous deep learning and clustering”. In: *international conference on machine learning*. 2017, pp. 3861–3870.

[13] Maziar Moradi Fard, Thibaut Thonet, and Eric Gaussier. “Deep k-means: Jointly clustering with k-means and learning representations”. In: *Pattern Recognition Letters* 138 (2020), pp. 185–192.

[14] Siddharth Krishna Kumar. “On weight initialization in deep neural networks”. In: *arXiv preprint arXiv:1704.08863* (2017).

[15] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[16] Han Xiao, Kashif Rasul, and Roland Vollgraf. “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms”. In: *arXiv preprint arXiv: 1708.07747* (2017).

[17] Jonathan J. Hull. “A database for handwritten text recognition research”. In: *IEEE Transactions on pattern analysis and machine intelligence* 16.5 (1994), pp. 550–554.

[18] Xinlei Chen and Deng Cai. “Large scale spectral clustering with landmark-based representation”. In: *25th AAAI conference on artificial intelligence*. 2011.

[19] Deng Cai et al. “Locality preserving nonnegative matrix factorization”. In: *Twenty-first international joint conference on artificial intelligence*. 2009.

[20] Yi Yang et al. “Image clustering using local discriminant models and global integration”. In: *IEEE Transactions on Image Processing* 19.10 (2010), pp. 2761–2773.

[21] Harold W Kuhn. “The Hungarian method for the assignment problem”. In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[22] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.