Deep Successive Subspace Learning for Data Clustering

1st Mohammadreza Sadeghi
Department of Electrical & Computer Engineering
McGill University
Mila - Québec AI Institute, Montreal, Canada
mohammadreza.sadeghi@mcgill.ca

2nd Narges Armanfard
Department of Electrical & Computer Engineering
McGill University
Mila - Québec AI Institute, Montreal, Canada
narges.armanfard@mcgill.ca

***Abstract*—Deep clustering combines embedding and clustering together to obtain an optimal low dimensional embedding subspace (aka latent subspace) for clustering, which can be more effective compared to conventional clustering approaches such as k-means. Typical deep clustering methods employ autoencoder (AE) and obtain their optimal latent space through minimizing data reconstruction loss which has no substantial connection with the clustering performance. In contrast, in this paper we propose a novel AE-based clustering scheme Deep Successive Subspace Learning (DSSL) which simultaneously minimizes weighted reconstruction and clustering losses of data points, where weights are defined based on similarity between latent representation of data points and cluster centers. DSSL obtains its optimal latent space through K (i.e. number of clusters) successive training runs where each run corresponds to an individual cluster. At each run, DSSL focuses on reconstruction and clustering of those data points that are more likely to belong to the corresponding cluster; hence, implicitly training those network parameters that have more influence on that cluster. Experimental results on benchmark datasets demonstrate that the proposed DSSL method can significantly outperform state-of-the-art clustering approaches.**

***Index Terms*—Deep Clustering, Autoencoders**

I. INTRODUCTION

In many science and engineering applications, the label information of data samples is non-observable or expensive to obtain. Clustering is an important data analysis tool in pattern analysis and machine learning. It strives to explore knowledge from unlabeled data. Many applications can be considered as typical examples of data clustering, such as the astronomical data analysis [1], the medical analysis [2], the gene sequencing [3], and the information retrieval [4], [5], [6], [7]. Clustering methods aim to group data points based on a similarity metric. Among different clustering methods, k-means [8] and fuzzy c-means [9] are the two popular conventional methods that are widely used in several applications [10, 11, 12], because of their simplicity. However, when data points are not evenly distributed around centroids, these algorithms fail to properly cluster data samples. Furthermore, they do not show good performance on high-dimensional data, while in many applications nowadays datasets are characterized by thousands of features [13]. Recently, deep learning-based clustering methods have been widely used in various applications such as image segmentation [14], social network analysis [15], face recognition [16], and computer vision [17]. The final goal of these methods is to find a new representation of data points in a lower dimensional space (aka subspace, latent space) which is more suitable for data clustering, e.g. by applying k-means to the obtained lower dimensional space. Applying autoencoders (AEs), which provide a highly non-linear transformation of data points, to the original data is the most common practice to find an optimal low-dimensional space in an unsupervised manner [18, 19, 20]. Encoder part of autoencoder projects the input data samples on a latent space, and decoder part endeavors to reconstruct the original input data using their latent representation; to this end, encoder and decoder networks are trained through minimizing reconstruction losses of input data.

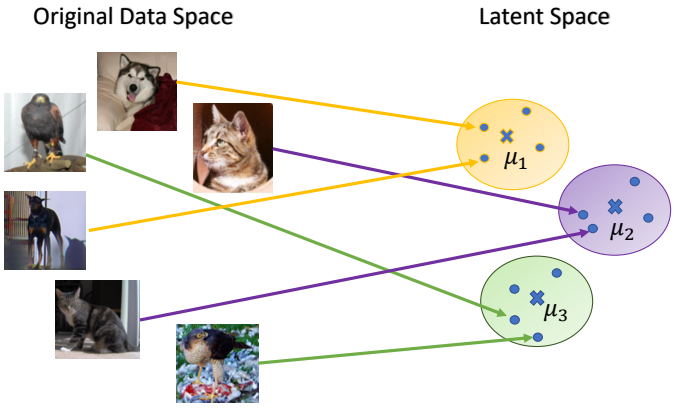

Fig. 1. The motivation of DSSL. Arrows show a nonlinear mapping from original input space to the latent space.

In this paper, we propose a deep successive subspace learning (DSSL) approach that jointly performs dimensionality reduction and clustering. DSSL is an AE-based network which is trained in an end-to-end manner. DSSL enhances clustering performance by incorporating weighted reconstruction and clustering losses, where weights are defined based on degree of similarity between latent representation of data points and cluster centers. As is discussed in Section II, there have been a few research studies that consider both reconstruction and clustering losses for data clustering [21, 22, 23]. The two main common disadvantages of the previous methods are: (1) at each training iteration, a data point is first assigned to a single

specific cluster (aka crisp assignment), then the clustering loss is computed over the clustered data, e.g. see [21, 22, 23]. A linear combination of clustering and reconstruction losses are then used to update the network parameters through a back-propagation process. Note that due to the unsupervised nature of the clustering problem, the *true* crisp assignment for each data point is unknown, and using an incorrect estimate of crisp assignments misleads the algorithm training phase. This issue becomes more critical when the non-crisp estimation of the K-dimensional cluster assignment vector (before snapping to 0 and 1) is far from the one-hot vector imposed by crisp assignment. (2) in all the previous methods, a single common loss function is used for all data clusters without considering the possible different characteristics of data in different clusters.

The proposed DSSL method addresses these two drawbacks by directly incorporating non-crisp (aka soft) assignments in the loss functions, where an individual loss function is designed to be minimized for each data cluster. The main contributions of DSSL lie in two folds:

- Parameters of the DSSL method are trained through K successive runs, where K is the number of clusters. The kth run, $k = 1, ..., K$, is associated to the kth cluster and minimizes its own distinct loss function – this encourages the DSSL network to focus on reconstruction and clustering of the data points that are more likely to belong to the kth cluster. In other words, at the kth run, the network is implicitly enforced to optimize those parameters that have more influence on the reconstruction and clustering of the kth cluster data.
- DSSL computes soft assignments for each data sample and employs them as samples weight when computing clustering loss at each of the K successive runs; this allows DSSL to take into account all possible cluster assignments when training its parameters. In addition, the soft assignments are incorporated in the DSSL's reconstruction loss. This allows the algorithm to focus on reconstructing those data that are more probable to belong to the kth cluster.

Block diagram of the proposed method's training phase is shown in Fig. 2.

II. Related Works

k-means [8] is the most popular clustering method that can be applied to a broad range of problems. Although this algorithm is fast and easy to implement, it does not show good results on high dimensional spaces. Also, it does not show good clustering performance when data points are not evenly distributed around their centroids in the original data points. To handle these difficulties, some algorithms such as [24, 25] perform subspace learning[1] (aka dimension reduction) to learn a low dimensional feature space (aka subspace, latent space) of data points and then employ the trained subspace for data clustering task. Another category of clustering algorithms, such as [28, 29], consider pairwise relationship between data points to embed the original high dimensional data points into a lower dimensional space and then apply k-means algorithm to the new space. These algorithms construct a weighted graph based on the relationships between the data points in the original space; they then solve an optimization problem based on Laplacian matrix of the graph. Although these methods outperform k-means, their computational cost for solving the optimization problem prevents their application when dealing with large datasets. Some studies, e.g. [30, 31], try to handle this problem by using stochastic optimization methods. For example, [30] formulates an adaptive stochastic gradient optimization, which is linear in the number of data that does not need storing the complete Laplacian matrix. Although the stochastic optimization based algorithm could outperform the original ones even in some small datasets, these methods only consider linear transformation of the data. In order to learn nonlinear transformations, which is crucial in clustering of more complicated datasets, [32] utilizes a deep autoencoder to get low dimensional feature space for a graph; it then applies the k-means algorithm to define clusters. Deep embedding network (DEN) [33] proposed an AE-based method that attempts to learn new representation of data points by enforcing group sparsity and locality-preserving constraints. In this algorithm, after finding the new representation, k-means algorithm is applied to define clusters. In order to enhance clustering performance, more recent algorithms jointly learn a new feature representation of the data and update cluster centers using the obtained new feature space. Deep embedding clustering (DEC) [34] uses a fully connected stacked autoencoder to initialize a new low dimensional feature space; after discarding the decoder part, students' t-distribution is used to assign data points to clusters and then an auxiliary target distribution is defined based on the cluster assignments. In the optimization phase of DEC, parameters of the encoder part and cluster centers are simultaneously updated to minimize the Kullback–Leibler (KL) divergence loss between the cluster assignments and the auxiliary target distribution, using stochastic gradient descent (SGD). There have been a few studies, e.g. [22, 21, 23], that aim to improve clustering performance through including both clustering and reconstruction errors in the loss function of their autoencoder. In order to preserve local structure of data, improved-DEC method (IDEC) [22] improves the DEC method by including the decoder part to add the reconstruction loss to the loss function of the original DEC method. In order to find a k-means friendly representation for data points, the loss function in the deep clustering network (DCN) method [21], which is a joint data dimensionality reduction and k-means clustering, consists of the k-means' objective and the reconstruction loss, where cluster centers are simultaneously updated using a discrete version of SGD. Deep k-means (DKM) [23] finds new representation and performs clustering at the same time by adding k-means loss function to the

[1]Note that there is another branch in data clustering called Subspace Clustering [26, 27], which is different from subspace learning. Subspace clustering techniques assume that data points are drawn from multiple subspaces corresponding to different data clusters.

reconstruction loss of an AE. SGD optimizer is used to update AE's weights and cluster centers.

III. Proposed Method

Consider a K-clustering problem with a dataset X consists of N samples, i.e. $X = \{x_1, \ldots, x_N\}$. The proposed DSSL network consists of an encoder and decoder. The encoder and decoder networks are respectively denoted by $f(.)$ and $g(.)$. Representation of X in the DSSL's latent space is shown by $U = \{u_1, u_2, ..., u_N\}$, where $u_j = f(x_j; \theta_e) \in R^d$ for $j = 1, ..., N$. θ_e denotes encoder parameters and d is the latent space dimension. The output of decoder is denoted by $\hat{x}_i = g(u_i; \theta_d)$, which represents the reconstructed input data x_i. Parameters of the decoder is denoted by θ_d. The kth cluster center is denoted by μ_k.

A. Training Phase

In order to initialize the DSSL's parameters, θ_e, θ_d and μ_k for $k = 1, \ldots, K$, we train an autoencoder using Adam optimization method [35], with the same parameters mentioned in the original paper, and back propagation algorithm. End-to-end training is performed by minimizing the mean squared error between input and output of the autoencoder, i.e. minimizing the reconstruction loss. We initialize θ_e, θ_d to the parameters of the trained autoencoder. We apply k-means algorithm to the latent space of the trained autoencoder and then initialize $\mu_k, k = 1, \ldots, K$ to the cluster centers defined by k-means.

In order to obtain a low dimensional latent space which is suitable for clustering, we propose to train the DSSL network with a novel loss function that is a weighted sum of reconstruction and clustering losses. We propose to train the DSSL network in K successive runs where at the kth run the DSSL network focuses on reconstructing and grouping of the data points that are more likely to belong to the kth cluster. In this way, at the kth run, we implicitly enforce the DSSL network to optimize those parameters that have more influence on the reconstruction and clustering of the kth cluster data.

Algorithm 1 DSSL method

Input: Data points X, number of clusters K, maximum iterations $MaxIter$, update interval T

Output: Crisp cluster assignment for query data

1: Initialize θ_e, θ_d and μ_k for $k = 1, \ldots, K$ (see Section III-A)
2: **for** $iter \in \{1, 2, ..., MaxIter\}$ **do**
3: **for** $k \in \{1, 2, ..., K\}$ **do**
4: Compute soft assignments p_{ik} using (2), for $i \in \mathfrak{B}$
5: Update DSSL parameters θ_e, θ_d, using SGD with momentum, employing loss function (1a)
6: **end for**
7: Every T iterations, update cluster centers using (3)
8: **end for**
9: Apply crisp assignment to the query data as is discussed in Section III-B

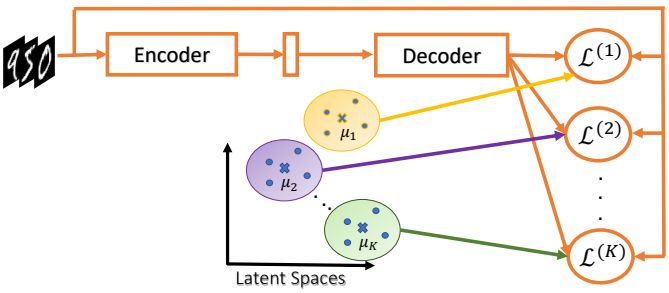

Fig. 2. Block diagram of DSSL's training procedure.

The proposed loss function at the kth run, $\mathcal{L}^{(k)}$, is shown in (1) where $\mathcal{L}_r^{(k)}$ and $\mathcal{L}_c^{(k)}$ respectively denote reconstruction and clustering losses. $\mathcal{L}_r^{(k)}$ ($\mathcal{L}_c^{(k)}$) consists of weighted reconstruction (clustering) losses of the data points where, to encourage focusing on the kth cluster samples, higher weights are assigned to the samples that are more "probable" to belong to the kth cluster. Probability of assigning data point x_i to the kth cluster is denoted by p_{ik}, which is also known as soft-assignment. Employing these soft (i.e. non-crisp) assignments in the clustering loss allows DSSL to take into account all possible cluster assignments when training its parameters. Incorporating soft assignments in the reconstruction loss allows the algorithm to focus on reconstructing those data that are more probable to belong to the kth cluster.

$$\mathcal{L}^{(k)} = \mathcal{L}_r^{(k)} + \zeta \mathcal{L}_c^{(k)} \tag{1a}$$

$$\mathcal{L}_r^{(k)} = \sum_{x_i \in \mathfrak{B}} p_{ik}^m ||x_i - \hat{x}_i||_2^2 \tag{1b}$$

$$\mathcal{L}_c^{(k)} = \sum_{x_i \in \mathfrak{B}} p_{ik}^m ||u_i - \mu_k||_2^2 \tag{1c}$$

In (1), ζ is a hyperparameter indicating the importance of the clustering loss in the networks training, and $\mathfrak{B}$ denotes a batch of data points.

In this paper, though other approaches are possible, we make use of the Fuzzy C-means to estimate p_{ik}, as is shown in (2), where m indicates the level of fuzziness and is set to 1.5 in all our experiments.

$$p_{ik} = \frac{\frac{1}{||u_i - \mu_k||_2^{2/(m-1)}}}{\sum_{j=1}^{K} \frac{1}{||u_i - \mu_j||_2^{2/(m-1)}}} \tag{2}$$

Finally, every T iterations, we update the cluster centers as is shown in (3).

$$\mu_k = \frac{\sum_{x_i \in X} p_{ik}^m u_i}{\sum_{x_i \in X} p_{ik}^m} \tag{3}$$

The pseudo code of the proposed DSSL method is presented in Algorithm 1.

B. Crisp assignment

We use the trained encoder and cluster centers to assign a cluster number to data points, i.e. crisp assignment. To this end, latent representation of the data points are computed using the encoder part of the DSSL network, then each data point is

TABLE I
ACC AND NMI (IN PARENTHESIS) ON THE BENCHMARK DATASETS FOR DIFFERENT CLUSTERING METHODS.

Method \ Datasets	MNIST	Fashion MNIST	2MNIST	CIFAR10	STL10
k-means	53.20 (50.00)	47.40 (51.20)	32.31 (44.00)	41.30 (38.14)	43.40 (34.56)
LSSC	71.40 (70.60)	49.60 (49.70)	39.77 (51.22)	50.31 (44.71)	61.50 (41.20)
LPMF	47.10 (45.20)	43.40 (42.50)	34.68 (38.69)	45.25 (34.11)	47.46 (29.14)
DEC	84.30 (83.72)	51.80 (54.63)	41.20 (53.12)	45.23(39.64)	45.12 (50.45)
IDEC	88.13 (83.81)	52.90 (55.70)	40.42 (53.56)	46.21(41.78)	47.84(53.27)
DCN	83.00 (81.00)	51.22 (55.47)	41.35 (46.89)	47.69 (41.23)	48.63 (56.21)
DKM	84.00 (81.54)	51.31 (55.57)	41.75 (46.58)	49.20 (40.56)	61.54 (63.59)
AE + k-means	86.03 (80.25)	57.94 (57.15)	41.23 (54.23)	55.94 (44.31)	62.20 (64.03)
DSSL	**92.17 (84.21)**	**59.11(62.05)**	**43.80 (55.52)**	**56.15 (45.30)**	**63.75 (64.51)**

TABLE II
ACC AND NMI (IN PARENTHESIS) ON IMBALANCED DATASETS FOR DIFFERENT CLUSTERING METHODS.

Method \ r	0.1	0.2	0.3	0.4	0.5
DEC	73.12 (67.89)	70.12 (61.72)	75.21 (68.32)	76.47 (71.25)	82.54 (73.22)
IDEC	72.28 (70.64)	78.91 (73.11)	82.55 (76.35)	83.55 (77.44)	84.22 (79.90)
DCN	72.74 (68.80)	73.02 (72.91)	79.60 (72.80)	74.52 (73.61)	76.32 (74.62)
DKM	45.96 (39.21)	46.21 (39.54)	48.56 (38.96)	51.25 (42.10)	50.98 (43.27)
AE + k-means	75.37 (68.12)	77.24 (72.31)	79.30 (74.20)	79.05 (73.60)	83.11 (74.83)
DSSL	**79.59 (72.00)**	**80.77 (74.65)**	**86.63(78.00)**	**86.30 (78.27)**	**88.61 (80.61)**

assigned to the cluster with nearest cluster center. Euclidean distance is used as the distance measure.

IV. EXPERIMENTS

Performance of the proposed method is demonstrated on various kind of datasets. Considering that clustering task is a fully unsupervised procedure, it is a common practice to concatenate train and test sets (e.g. See [34, 22, 21, 23]). The datasets are: MNIST [36] which contains 60,000 training and 10,000 testing samples, each being a 28×28 handwritten digit monochrome image; Fashion MNIST [37] that has the same number of images and the same image size with MNIST, but it is fairly more complicated since instead of digits, Fashion MNIST consists of various types of fashion products; CIFAR-10 consists of 60,000 RGB images of 10 different objects where the size of each image is 32×32; STL-10 is similar to CIFAR-10, which comprise of 13,000 96×96 RGB images from 10 classes; 2MNIST, which is a more challenging dataset coming from two different distributions, is created by combining the two datasets MNIST-full and Fashion MNIST and hence has 140,000 images of size 28×28 with 20 classes. We use a pre-trained VGG-16 network on Imagenet [38] to extract the useful features from RGB images, i.e. CIFAR10 and STL-10 datasets.

For the grey scale datasets MNIST, Fashion MNIST, and 2MNIST, following [39], we use a symmetric structure of convolutional AE, which can take one image channel as its input. For the RGB datasets CIFAR-10 and STL-10, following [34], after applying the pre-trained VGG-16 network, we use fully connected networks in forming our DSSL network. All experiments are performed in Google Colaboratory.

Clustering performance of the proposed method is compared against several well-known and state-of-the-art clustering methods. The comparison algorithms include non-deep learning methods: k-means [8], large-scale spectral clustering (LSSC) [40] and locality preserving non-negative matrix factorization (LPMF) [41], as well as deep learning-based algorithms: DEC [34], IDEC [22], DCN [21] and DKM [23]. In addition, in order to demonstrate the effectiveness of the proposed DSSL method, we report the results on our baseline method, AE + k-means. In AE + k-means, k-means is applied to the latent space of a single regular AE trained with the corresponding structure discussed above.

A. Evaluation metrics

To evaluate the clustering performance, we use two standard evaluation metrics: clustering accuracy (ACC) [42] and normalized mutual information (NMI) [43], as are shown below.

$$ACC = \max_m \frac{\sum_{i=1}^{N} 1\{l_i = m(c_i)\}}{N} \tag{4a}$$

$$NMI = \frac{I(l;c)}{max\{H(l), H(c)\}} \tag{4b}$$

ACC finds the best match between the true labels and the predicted cluster assignments. NMI computes the normalized measure of similarity between two labels of the same data point. In (4), l_i and c_i are respectively the ground truth label and the cluster assignment (i.e. predicted label) for data point x_i, mutual information between the true labels $l = \{l_1, \ldots, l_N\}$ and the predicted labels $c = \{c_1, \ldots, c_N\}$ is denoted by $I(l; c)$, $H(.)$ represents the entropy function, and 1 denotes the indicator function. NMI and ACC lie in the range of 0 to 1, with 0 being the worst clustering result and 1 the perfect performance.

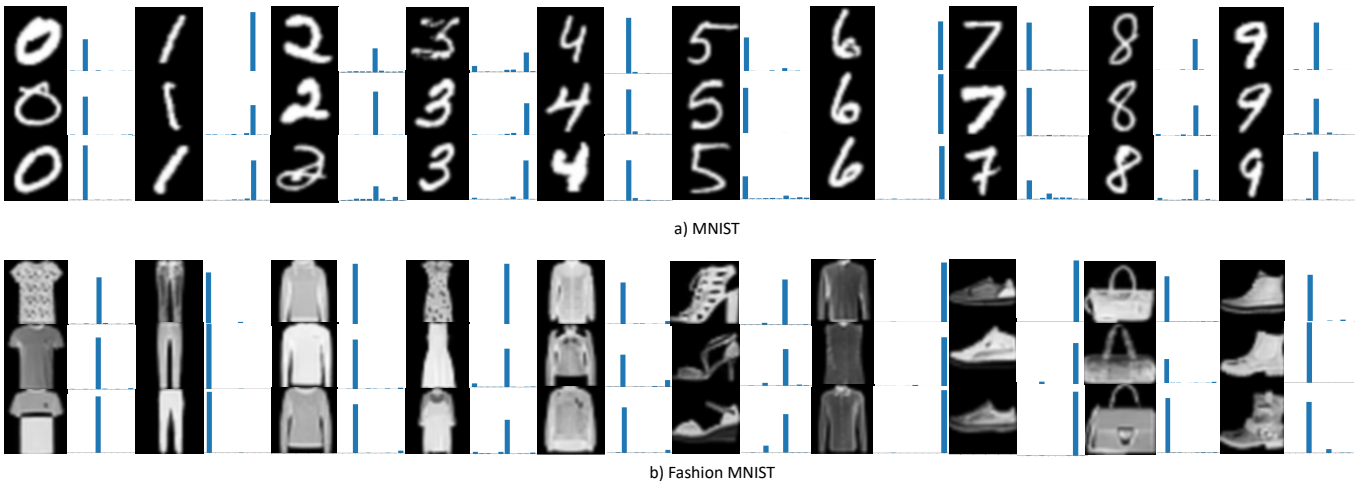

Fig. 3. Soft assignments for some samples of MNIST and Fashion MNIST datasets. Y axis ranges between 0 and 1.

B. Clustering performance

Clustering performance of our comparison methods and the proposed DSSL method, are shown in Table I. For the comparison methods, we run the code released by the corresponding authors where we used the same hyper-parameters as is mentioned in the original papers. The best result for each dataset is shown in bold. Among the nine algorithms, the proposed DSSL method yields the best results on all reported clustering performances; on average, over the five datasets, DSSL improves ACC and NMI of the comparison deep-learning based methods by 10.67% and 14.38%, respectively. In addition, it can be seen that DSSL significantly improves the clustering performance of our baseline method AE + k-means, which verifies the big advantage of the proposed method over the traditional AE-based clustering approaches; on average, over the five datasets, DSSL improves ACC and NMI of our baseline AE + k-means respectively by 2.33% and 2.32% .

C. Performance on imbalanced datasets

To show the effectiveness of the proposed framework on imbalanced data, we sample subsets of MNIST with various retention rates $r \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, where data points of class 0 are kept with probability r and class 9 with probability 1, with the other classes linearly in between. As such, the smallest cluster is r times smaller than the largest cluster. ACC and NMI for various r on all datasets are shown in Table II. It can be seen that the proposed DSSL method significantly outperforms our comparison methods for all r values; e.g. DSSL is more than 3.57% more accurate than the best comparison method, for $r = 0.4$ and 0.5. On average, over the five imbalanced datasets, DSSL has improved AE + k-means by 5.57% in ACC and 4.09% in NMI.

D. Soft assignments visualization

In Fig. 3, we visualize soft assignments learned by DSSL on some samples from the two benchmark datasets MNIST and Fashion MNIST. As is expected, we observe that if two samples belong to the same cluster, the same index takes the highest value in the corresponding K-dimensional soft assignment vector. This demonstrates the DSSL abilities in finding similarities between data points in an unsupervised manner, even for noisy corrupted samples such as the last figure in column 3 (from left) of Fig. 3 (a).

E. Effect of number of data samples

In order to see impact of number of data samples to the proposed DSSL and other deep-learning based methods, we vary number of samples for CIFAR-10 between 10,000 and 60,000 with an interval of 10,000. Fig 4 shows that the performance (ACC and NMI) of all methods increases when more data samples are provided. This implies that more data samples are beneficial for data clustering. In addition, we observe that, DSSL maintains higher performance at all points, compare to our comparison clustering methods; this can be assigned to the fact that DSSL considers a customized loss function for every cluster, through employing soft assignments.

F. t-SNE visualization

In Fig 5, the effectiveness of our proposed DSSL method is further demonstrated by comparing different representation of MNIST using t-SNE [44], where the learned representation of each algorithm is mapped to a 2-dimensional space. As it can be seen, clusters indicated by DSSL are more clearly scattered around cluster centers compare to other deep-learning based

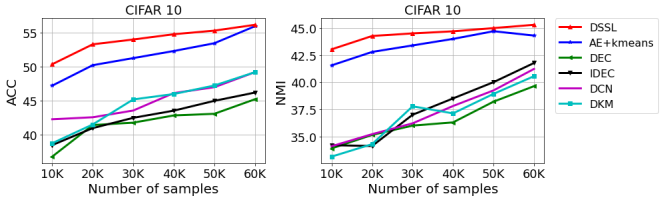

Fig. 4. Effect of number of samples on clustering performance for different methods, on CIFAR-10 dataset.

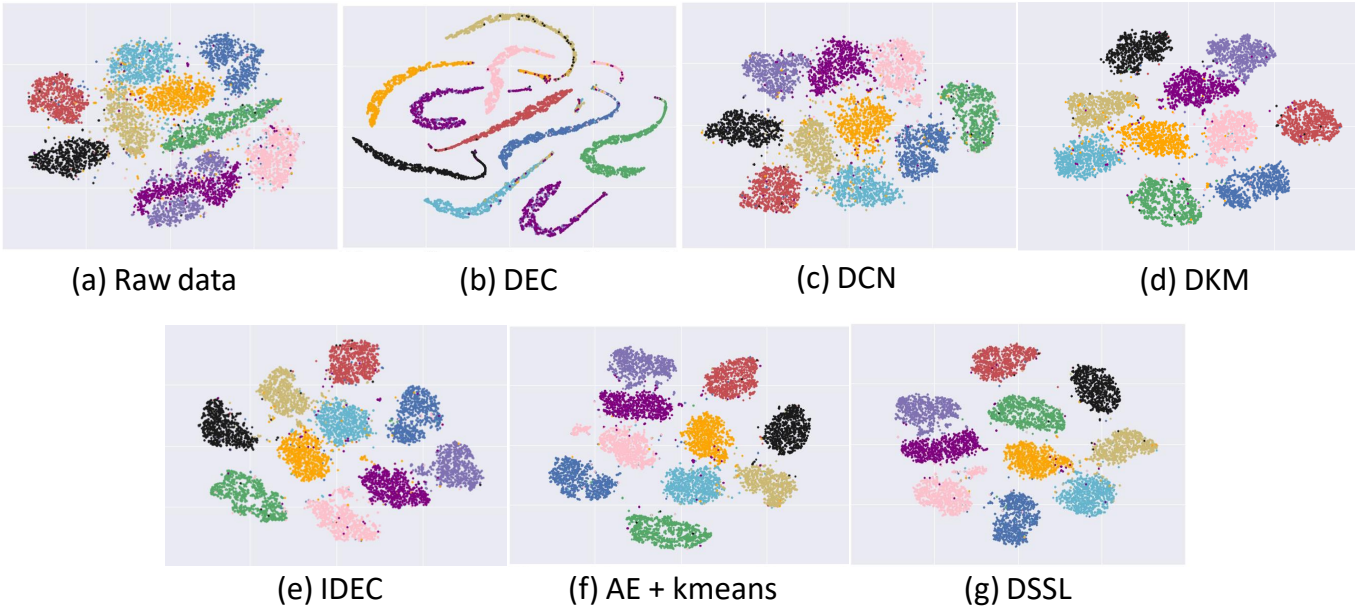

Fig. 5. Visualization of different methods using t-SNE. Axes range from -100 to 100.

clustering methods. The improved performance of DSSL is more noticeable when considering separation of groups colored in magenta and purple (digits 4 and 9), as well as the separation of clusters in cyan, olive and orange (digits 3, 5 and 8).

G. Hyperparameters

The proposed DSSL method was implemented in python using the deep-learning framework Pytorch. We set maximum number of training epochs for all experiments to 100 and used stochastic gradient descent for optimizing our proposed loss function. In all experiments, the hyperparameters ζ, d and m are respectively set to 0.1, 10 and 1.5. The update interval parameter T is set to 2 in all experiments. The number of clusters, K, is a user-defined parameter and is set to its true value for each dataset, for all algorithms. If K is unknown, one may use [45] to estimate it.

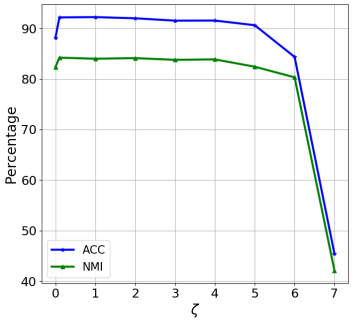

Fig. 6. ACC and NMI of the proposed DSSL method for dataset MNIST where the parameter ζ ranges from 0 to 7.

We investigate the effect of changing ζ on the performance of DSSL, for the MNIST dataset, where ζ lies between 0 and 7. As it is shown in Figure 6, by increasing ζ from 0 to 0.1, the DSSL performance significantly improves in both NMI and ACC. DSSL maintains high ACC and NMI for ζ between 0.1 and 5. It demonstrates that the DSSL method is not too sensitive to a wide range of values of ζ, as is desired.

Moreover, we scrutinize the effect of the update interval hyperparameter T in the clustering performance of the proposed DSSL method, where $T \in \{2, 5, 10, 25, 50\}$. Fig 7 shows ACC and NMI vs. epochs for the different T values, for the MNIST dataset. As is expected, better ACC and NMI are achieved for shorter update intervals, i.e. for smaller T values.

Fig. 8 shows the effect of level of fuzziness hyperparameter m in the DSSL clustering performance, where $m \in \{1.1, 1.3, 1.5, 1.7\}$. When m $\rightarrow$ 1 (m $\rightarrow \infty$), the soft assignments vectors converge to one-hot (equal-probability) vectors. As it is shown in Fig. 8, the best performance, on MNIST, is obtained for $m = 1.5$.

V. Conclusion

In this study, we propose a novel and effective AE-based deep-clustering method DSSL that simultaneously embeds

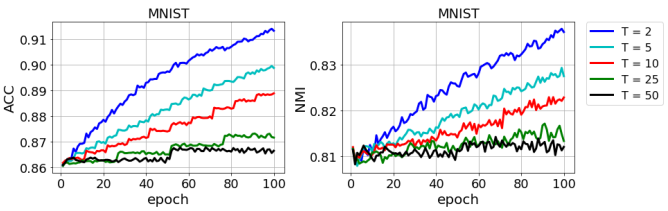

Fig. 7. Effect of update interval T on clustering performance.

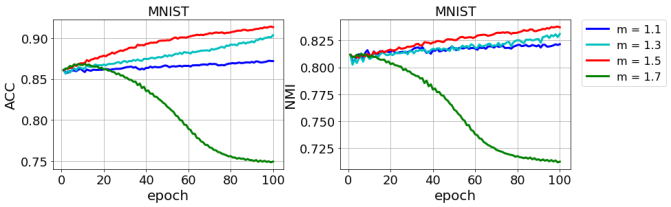

Fig. 8. Effect of level of fuzziness m on clustering performance.

data points into a lower dimensional space and assigns data points to their corresponding clusters. Unlike most deep-learning based clustering algorithms that consider crisp assignments (which are unknown at the problem outset) through their learning process, we propose to employ soft assignments; i.e., we propose to minimize weighted reconstruction and clustering losses, where weights are determined based on the degree of similarity between low dimensional representation of data points and cluster centers. Moreover, we proposed to train parameters of the DSSL network through K successive iterations where each iteration is corresponded to a specific cluster. At each iteration, by minimizing the proposed loss function, we encourage the DSSL network to concentrate on reconstructing and clustering of a portion of data points that are more likely to be in the corresponding cluster. Effectiveness of the DSSL method is demonstrated on benchmark datasets through an extensive set of experiments.

REFERENCES

[1] Ashley J Ross et al. "The clustering of the SDSS DR7 main Galaxy sample–I. A 4 per cent distance measure at z= 0.15". In: *Monthly Notices of the Royal Astronomical Society* 449.1 (2015), pp. 835–847.

[2] Nguyen Dang Thanh, Mumtaz Ali, et al. "Neutrosophic recommender system for medical diagnosis based on algebraic similarity measure and clustering". In: *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE. 2017, pp. 1–6.

[3] S Selva Kumar and H Hannah Inbarani. "Analysis of mixed C-means clustering approach for brain tumour gene expression data". In: *International Journal of Data Analysis Techniques and Strategies* 5.2 (2013), pp. 214–228.

[4] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823.

[5] Kaiye Wang et al. "Joint feature selection and subspace learning for cross-modal retrieval". In: *IEEE transactions on pattern analysis and machine intelligence* 38.10 (2015), pp. 2010–2023.

[6] Linan Feng and Bir Bhanu. "Semantic concept co-occurrence patterns for image annotation and retrieval". In: *IEEE transactions on pattern analysis and machine intelligence* 38.4 (2015), pp. 785–799.

[7] Syed Sameed Husain and Miroslaw Bober. "Improving large-scale image retrieval through robust aggregation of local descriptors". In: *IEEE transactions on pattern analysis and machine intelligence* 39.9 (2016), pp. 1783–1796.

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

[9] 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.

[10] 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.

[11] 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).

[12] 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.

[13] M Pavithra and R Parvathi. "A survey on clustering high dimensional data techniques". In: *International Journal of Applied Engineering Research* 12.11 (2017), pp. 2893–2899.

[14] 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.

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

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

[17] 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.

[18] Seunghyoung Ryu et al. "Convolutional autoencoder based feature extraction and clustering for customer load analysis". In: *IEEE Transactions on Power Systems* 35.2 (2019), pp. 1048–1060.

[19] Chunfeng Song et al. "Auto-encoder based data clustering". In: *Iberoamerican congress on pattern recognition*. Springer. 2013, pp. 117–124.

[20] Pierre Baldi. "Autoencoders, unsupervised learning, and deep architectures". In: *Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings. 2012, pp. 37–49.

[21] Bo Yang et al. "Towards k-means-friendly spaces: Simultaneous deep learning and clustering". In: *interna-*

tional conference on machine learning. 2017, pp. 3861–3870.

[22] Xifeng Guo et al. "Improved deep embedded clustering with local structure preservation." In: *International Joint Conference on Artificial Intelligence(IJCAI)*. 2017, pp. 1753–1759.

[23] 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.

[24] Irem Ersöz Kaya et al. "PCA based clustering for brain tumor segmentation of T1w MRI images". In: *Computer methods and programs in biomedicine* 140 (2017), pp. 19–28.

[25] Kewei Tang et al. "Robust subspace learning-based low-rank representation for manifold clustering". In: *Neural Computing and Applications* 31.11 (2019), pp. 7921–7933.

[26] Bhagyashri A Kelkar and Sunil F Rodd. "Subspace clustering—A survey". In: *Data Management, Analytics and Innovation*. Springer, 2019, pp. 209–220.

[27] Lance Parsons, Ehtesham Haque, and Huan Liu. "Subspace clustering for high dimensional data: a review". In: *Acm sigkdd explorations newsletter* 6.1 (2004), pp. 90–105.

[28] Andrew Y Ng, Michael I Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm". In: *Advances in neural information processing systems*. 2002, pp. 849–856.

[29] Santo Fortunato. "Community detection in graphs". In: *Physics reports* 486.3-5 (2010), pp. 75–174.

[30] Y. Han and M. Filippone. "Mini-batch spectral clustering". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017, pp. 3888–3895.

[31] M. El Gheche, G. Chierchia, and P. Frossard. "Stochastic Gradient Descent for Spectral Embedding with Implicit Orthogonality Constraint". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 3567–3571.

[32] Fei Tian et al. "Learning deep representations for graph clustering". In: *28th AAAI Conference on Artificial Intelligence*. 2014.

[33] Peihao Huang et al. "Deep embedding network for clustering". In: *2014 22nd International conference on pattern recognition*. IEEE. 2014, pp. 1532–1537.

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

[35] Diederik P Kingma and Jimmy Ba. "Adam: a method for stochastic optimization". In: *arXiv preprint arXiv: 1412.6980* (2014).

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

[37] 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).

[38] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE. 2009, pp. 248–255.

[39] Yuan Xie et al. "Joint Deep Multi-View Learning for Image Clustering". In: *IEEE Transactions on Knowledge and Data Engineering* (2020).

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

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

[42] 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.

[43] Wei Xu, Xin Liu, and Yihong Gong. "Document clustering based on non-negative matrix factorization". In: *26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. 2003, pp. 267–273.

[44] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[45] Chunhui Yuan and Haitao Yang. "Research on K-value selection method of K-means clustering algorithm". In: *J—Multidisciplinary Scientific Journal* 2.2 (2019), pp. 226–235.