



Maintaining Consistency with Constraints: A Constrained Deep Clustering Method

Yi Cui, Xianchao Zhang^(✉), Linlin Zong, and Jie Mu

School of Software, Dalian University of Technology, Dalian 116620, China

Abstract. Constrained clustering has been intensively explored in the data mining. Popular clustering algorithms such as k -means and spectral clustering are combined with prior knowledge to guide the clustering process. Recently, constrained clustering with deep neural network gains superior performance by jointly learning cluster-oriented feature representations and cluster assignments simultaneously. However, these methods face a common issue that they have poor performance when only minimal constraints are available because of their single way to mine constraint information. In this paper, we propose an end-to-end clustering method that learns unsupervised information and constraint information in two consecutive modules: an unsupervised clustering module to obtain feature representations and cluster assignments followed by a constrained clustering module to tune them. The constrained clustering module is composed of a Siamese or triplet network to maintain consistency with constraints. To capture more information from minimal constraints, the consistency is maintained from two perspective simultaneously: embedding space distance and cluster assignments. Extensive experiments on both pairwise and triplet constrained clustering validate the effectiveness of the proposed algorithm.

Keywords: Constrained clustering · Semi-supervised clustering · Deep clustering · Metric learning

1 Introduction

Clustering with deep neural networks has extensively explored due to the inherent property of highly non-linear transformation of DNNs. These methods effectively combine the neural network with popular clustering algorithms, such as k -means [7, 14, 22], spectral clustering [17], subspace clustering [10], agglomerative clustering [23] to joint dimensionality reduction and clustering-oriented representation learning. These unsupervised methods refer to unlabeled data, however, some prior knowledge such as pairwise constraints or triplet constraints could be obtained automatically in many clustering tasks.

This work was supported by National Science Foundation of China (No.61632019; No.61876028; No.61972065; No.61806034).

Constrained clustering is a kind of task that few auxiliary information is provided to guide clustering. Some constrained clustering methods are explored with pairwise constraints (must-link and cannot-link) [8, 16]. SDEC [16] decreases the embedding distance between must-link pairs and increases distance between cannot-link pairs. But the distance in the embedding space between cannot-link pairs have already been large at the beginning of training due to the good separation of the pre-trained network, which leads to the inefficiency of its objective. Hsu et al. [8] present their objective on softmax output with KL divergence but abandon the contribution of instances without constraints. Zhang et al. [25] explore more complex constraints. They enforce the must-link pairs with similar assignment probability and cannot-link pairs oppositely. But when the number of constraints is not enough to mitigate the negative effect of imbalance (which means very few must-link assignments can be referred to, e.g. approximately 10% in Fashion dataset), this method that only mines constraint information from the perspective of cluster assignments is sensitive to the reduction of the number of constraints. For these reasons, these methods face a common issue that they have poor performance when the number of constraints is small.

In this paper, we propose a Constrained Deep Clustering method (CDC) that aims to maintain consistency with constraints. To be effective even if minimal constraints are available, our method learns unsupervised information and constraint information in two consecutive modules: an unsupervised clustering module followed by a constrained clustering module. Inspired by the metric learning, we construct the network based on a Siamese network or triplet network in the constrained clustering module. For the purpose of capturing more information from minimal constraints, the consistency is maintained from two perspective simultaneously: embedding space distance and cluster assignments. The model is trained by cosine function as the similarity metric avoiding the inefficiency when embedding distance between cannot-link pairs is large and weighted cross entropy objective to tune cluster assignments. The main contributions of this paper are summarized as follows:

- We propose an end-to-end clustering method that learns unsupervised information and constraint information in two consecutive modules: an unsupervised clustering module to obtain feature representations and cluster assignments followed by a constrained clustering module to tune them.
- We propose effective objective function to maintain consistency with constraints from two perspective: embedding space distance and cluster assignments.
- Extensive experiments are conducted on both image and text datasets. The results show competitive performance on both pairwise and triplet constrained clustering, validating the effectiveness of CDC algorithm.

2 Related Work

Deep clustering is a category of clustering in recent years that combine deep neural network to learn cluster-friendly features. There are approaches [6, 7, 21, 22] obtaining feasible feature space based on autoencoder (AE). Other novel

methods adopt deep generative model to perform clustering task, such as VAE-based [5, 11] and GAN-based [3, 15, 24] methods. In addition, some clustering methods recently has shifted to handle high-dimensional data, including spectral clustering [9, 17] and subspace clustering [10, 26, 27].

Constrained clustering has been widely studied to lead an auxiliary guidance to clustering. Some methods explore strategies for improving clustering performance with pairwise constraints [1, 2, 18, 19]. Other methods with deep neural network gains better performance. Hsu et al. [8] view the outputs of the softmax layer as the distribution of possible clusters given a sample and evaluate the similarity with KL divergence. Zhang et al. [25] explore more complex constraints generated from new types of side information. Although these methods capture the point that similar samples should output similar assignment distribution, there is no work noticing consistency of embedding space distance and cluster assignments simultaneously.

3 Proposed Method

Consider a task about clustering a data set X containing n unlabeled instances, each sample $\{x_i \in \mathbb{R}^d\}_{i=1}^n$ should be assigned to one of k clusters. Except these unlabeled data, two types of user-specified prior information is also provided to guide the clustering process, including pairwise constraints and triplet constraints. A pairwise constraint indicates that a pair of samples $\{(x_i, x_j) : x_i, x_j \in X\}$ have a relationship of must-link (x_i and x_j belong to the same clusters) or cannot-link (x_i and x_j belong to different clusters). A triplet constraint consists of a triple of samples $\{(\tilde{x}, x_p, x_n) : \tilde{x}, x_p, x_n \in X\}$, where the positive sample x_p is closer to the anchor \tilde{x} than the negative sample x_n in the embedding space.

We propose to find a non-linear mapping $f_\theta : X \rightarrow Z$ that transforms the original data into latent space Z , in which the embedding distance is consistent with the original semantic distance and cluster assignments are consistent with constraints. The model contains two consecutive modules: the unsupervised clustering module followed by our constrained clustering module. The whole structure of CDC is illustrated in Fig. 1.

We introduce the referred method in unsupervised clustering module in Sect. 3.1. Then we propose two types of constrained clustering module with pairwise constraints and triplet constraints respectively in Sect. 3.2 and Sect. 3.3.

3.1 Unsupervised Clustering Module

The first module aims to learn cluster-oriented feature representations. We refer the DEC [21] to learn feature representations and cluster assignments.

The DEC method initializes the centroids $\{\mu_j\}_{j=1}^k$ through k -means on the embedding space of the autoencoder pre-trained by a stacked autoencoder (SAE), then computes the soft assignments q_{ij} as:

$$q_{ij} = \frac{(1 + \|z_i - \mu_j\|^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'=1}^k (1 + \|z_i - \mu_{j'}\|^2/\alpha)^{-\frac{\alpha+1}{2}}}, \quad (1)$$

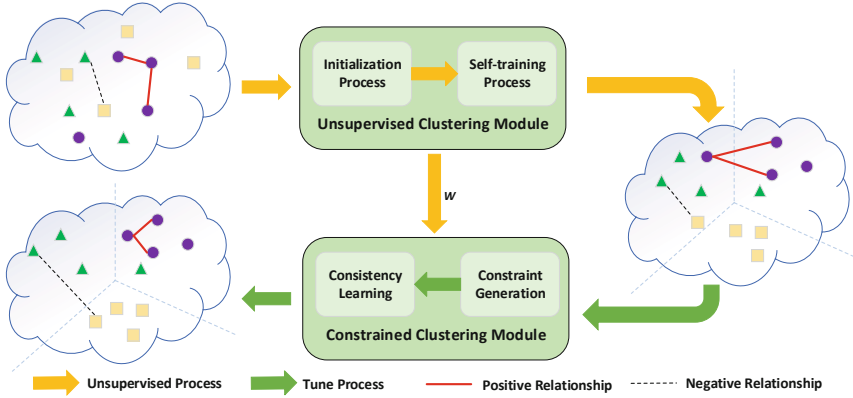


Fig. 1. The process of CDC algorithm. The method learns unsupervised information and constraint information in two consecutive modules: an unsupervised clustering module to obtain feature representations and cluster assignments followed by a constrained clustering module to tune them.

where q_{ij} measures the similarity between embedded data z_i and centroids μ_j with Student’s t-distribution being the kernel, α is a constant, e.g. $\alpha = 1$.

The auxiliary distribution P is defined to refine the cluster assignments. By squaring the soft assignments q_{ij} and then normalizing it, p_{ij} is formulated as:

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_{j'=1}^k (q_{ij'}^2 / \sum_i q_{ij'})}. \tag{2}$$

The loss function is defined as the reconstruction loss added to the KL divergence between soft assignments Q and auxiliary distribution P as follows:

$$L = KL(P||Q) + L_R = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} + \sum_i \|x_i - x'_i\|^2. \tag{3}$$

The clusters are iteratively refined during this self-training process. Constrained clustering module inherits the parameters and centroids and then learn from pairwise constraints or triplet constraints.

3.2 Clustering with Pairwise Constraints

The pairwise constraints are learned in our constrained clustering module based on a Siamese architecture, which is a popular network in metric learning. Two samples with pairwise constraints are required as inputs at the same step. Each group of inputs can be expressed as a triad $((x_1, x_2), y)$, where y is an indicator that $y = 1$ when given x_1 and x_2 with must-link relationship while $y = 0$ with cannot-link constraint. The structure of pairwise constrained clustering module is illustrated in Fig. 2. For the purpose of maintaining consistency with constraints, we define the objective function in two parts: embedding space distance and cluster assignments.

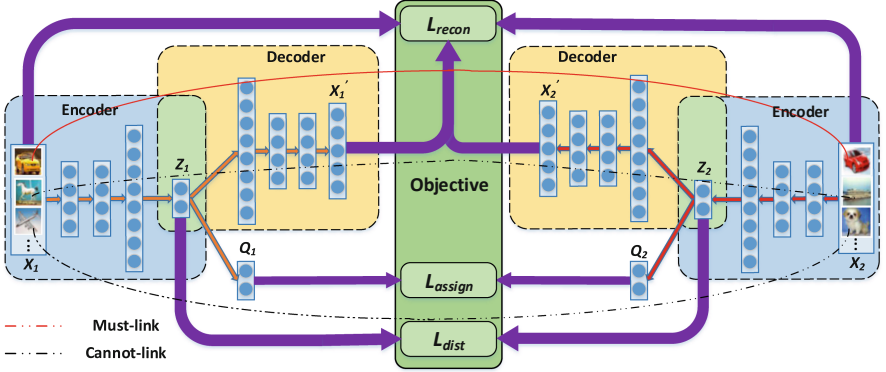


Fig. 2. The structure of constrained clustering module on pairwise constrained clustering based on a Siamese network. Constrained pairs are transformed into embedded features Z_1 and Z_2 . Soft assignments Q_1 and Q_2 are normalized to compute assignment objective. The shared parameters are optimized by Eq. (7).

Consistency of Embedding Space Distance. The main idea of this part is to seek a mapping that transforms pairs of inputs into a embedding space, in which a similarity measure approximates the semantic information in the original space. To this end, the distance loss for all m groups of $((x_1, x_2), y)$ is defined as:

$$L_{dist} = -\frac{1}{m} \sum_{i=1}^m \left(y^{(i)} \lambda_1 \sigma(z_1^{(i)}, z_2^{(i)}) - (1 - y^{(i)}) \lambda_2 \sigma(z_1^{(i)}, z_2^{(i)}) \right), \quad (4)$$

where $z_1^{(i)}$ and $z_2^{(i)}$ are corresponding embedded features of the i^{th} group of inputs, $\sigma(\cdot)$ is a similarity function, λ_1 and λ_2 are trade-off parameters. In summary, the embedded features with the same label prefer larger similarity, while points with different labels obtain smaller similarity by minimizing the objective function.

Consistency of Cluster Assignments. The main idea of this part is to tune cluster assignments with given constraints. Soft assignments are learned from its high confidence assignments in the unsupervised clustering module. We expect to tune cluster assignments to maintain the consistency with constraints. Specifically, must-link pairs are expected to have similar cluster assignments distribution, while assignment differences of cannot-link pairs are strengthened. The assignment loss is formulated as:

$$L_{assign} = -\frac{1}{m} \sum_{i=1}^m \left(y^{(i)} \lambda_3 w^{(i)} \log(q_1^{(i)} \cdot q_2^{(i)}) + (1 - y^{(i)}) w^{(i)} \log(1 - q_1^{(i)} \cdot q_2^{(i)}) \right). \quad (5)$$

This process is treated as a binary classification problem that whether or not two constrained samples belong to the same cluster. The inner product of corresponding normalized soft assignments $q_1^{(i)}$ and $q_2^{(i)}$ reflects the probability that

two inputs $x_1^{(i)}$ and $x_2^{(i)}$ are assigned into the same cluster. By minimizing the cross entropy loss, the must-link pairs prefer to be allocated into the same cluster and the cannot-link pairs are the opposite. In addition, we introduce a weight w to pay more attention to those pairs whose distances in the embedding space are not consist with constraints. Precisely speaking, the weights increase for those must-link pairs with large differences in embedded features and those cannot-link pairs with small differences. The weight formulas are defined as:

$$w^{(i)} = \begin{cases} \frac{1}{1 + e^{-d^{(i)}}}, & \text{if } (x_1, x_2)^{(i)} \in \text{must-link}, \\ \frac{3 + e^{d^{(i)}}}{2(1 + e^{d^{(i)}})}, & \text{if } (x_1, x_2)^{(i)} \in \text{cannot-link}, \end{cases} \quad (6)$$

where $d^{(i)} = \alpha \|z_1^{(i)} - z_2^{(i)}\|_2$ reflects the difference between a pair of embedded features, α is an adjustment parameter to control the distance. We set $\alpha = 0.01$ in all experiments because the great masses of samples are well-separated. The weight w is a monotonically increasing function for must-link, while monotonically decreasing function in the opposite case.

In summary, we define the objective function in constrained clustering module for pairwise constraints as:

$$L_{pair} = L_{dist} + L_{assign} + L_{recon}, \quad (7)$$

$$L_{recon} = \frac{1}{m} \sum_{i=1}^m y^{(i)} (\|x_1 - x'_1\|^2 + \|x_2 - x'_2\|^2)^{(i)}, \quad (8)$$

where L_{recon} is the sum of reconstruction losses of two instances, which is added to the must-link cases to avoid a large scale cluster.

3.3 Clustering with Triplet Constraints

Triplet constraints are weaker constraints and easily accessible with only a trained embedding space. They could replace the stronger constraints in some constrained clustering tasks that lack ground truth labels or partition-based constraints, e.g. pairwise constraints. Different from these stronger constraints coming from specific partitions, triplet constraints convey the differences in distance level.

We construct a triplet network for training triplet constraints. As we can see in Fig. 3, a triple of samples (\tilde{x}, x_p, x_n) are input to the network simultaneously. The similarities $\sigma(\tilde{z}, z_n)$ and $\sigma(\tilde{z}, z_p)$ are calculated in the embedding space output by the network with shared parameters. The objective function in constrained clustering module for triplet constraints is formulated as:

$$L_{trip} = \max(\sigma(\tilde{z}, z_n) - \sigma(\tilde{z}, z_p) + m, 0), \quad (9)$$

where $\sigma(\tilde{z}, z_n)$ and $\sigma(\tilde{z}, z_p)$ represent similarities between positive and negative samples against the anchor respectively. Those positive samples are pulled close

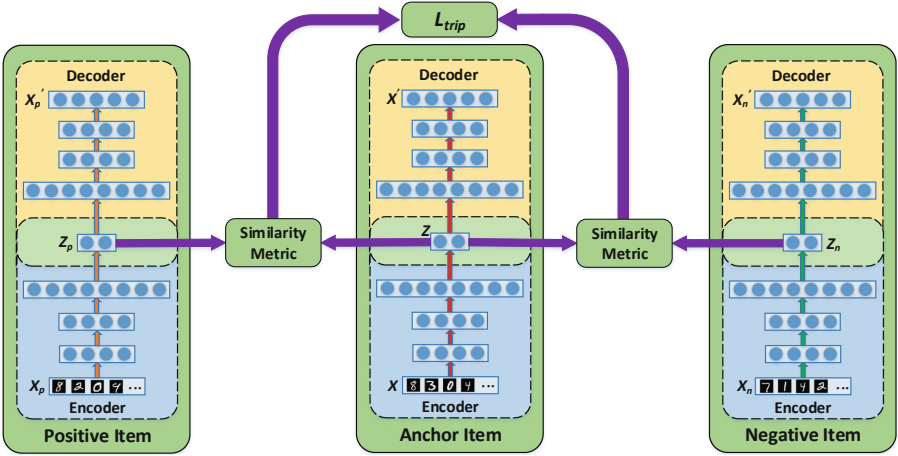


Fig. 3. The structure of constrained clustering module on triplet constrained clustering based on a triplet network. A triple of samples are input into the network at the same step. The similarities are obtained in the embedding space. Parameters are shared among the triplet network and are optimized by Eq. (9).

to their anchor and negative samples are separated from them. A hyperparameter margin m is introduced as a threshold that tries to widen the gap in $\sigma(\tilde{z}, z_n)$ and $\sigma(\tilde{z}, z_p)$. Due to the partition uncertainty of triplet constraints, some cases cannot be avoided that some positive samples and their anchors come from different classes, or some negative samples have the same labels with their anchors, which we call *imperfect triplet constraints*. The margin m also works by preventing x_p being too close or x_n being too separated from \tilde{x} in these cases. The parameter study about m is illustrated in Sect. 4.4.

In summary, our method learns feature representations and cluster assignments in the unsupervised clustering module and then tunes them in the constrained clustering module in one epoch. The procedure is summarized in Algorithm 1.

4 Experiments

4.1 Datasets

To verify the effectiveness and efficiency of the proposed CDC on constrained clustering tasks, we evaluate it on five benchmark datasets:

- **MNIST** [12]: A dataset composed of 70000 handwritten digits of 10 types. Each sample is a 28×28 gray image.
- **Fashion-MNIST** [20]: A dataset of Zalando’s article images with the same size as MNIST. Each sample is a 28×28 gray image, divided into 10 classes.
- **USPS**: A handwritten digits dataset that contains 9298 images (7291 for training, 2007 for test) with size of 16×16 pixels.

Algorithm 1. Constrained Deep Clustering (CDC)

Input: Dataset X , pairwise or triplet constraint dataset \tilde{X} , number of clusters k .
Output: Embedded features Z and cluster assignment vector s .
Initialization: Pre-train the stacked denoising autoencoder layer by layer to obtain Z . Obtain k initial centers $\{\mu_j\}_{j=1}^k$ with k -means in space Z .

- 1: **while** not reach the maximum epochs **do**
- 2: **Unsupervised clustering module:**
- 3: **for** every mini-batch data in X **do**
- 4: Obtain $z_i = f_\theta(x_i)$ through the encoder.
- 5: Compute q_{ij} and p_{ij} according to Eq. (1, 2).
- 6: Update θ and $\{\mu_j\}_{j=1}^k$ by minimizing Eq. (3).
- 7: **Constrained clustering module:**
- 8: **for** every mini-batch data in \tilde{X} **do**
- 9: Obtain (z_1, z_2) or (\tilde{z}, z_p, z_n) through Siamese or triplet network.
- 10: Update θ and $\{\mu_j\}_{j=1}^k$ by minimizing pairwise loss or triplet loss Eq. (7, 9).
- 11: Obtain Z and $s_i = \arg \max_j q_{ij}$ for all instances.
- 12: **if** stopping criterion is met **then**
- 13: Stop training.

- **KMNIST** [4]: Kuzushiji-MNIST is a dataset which focuses on cursive Japanese, composed of 28×28 images of 10 types. Train and test set sizes are 6,000 and 1,000 per class.
- **Reuters10K** [13]: A subset consist of 10000 examples of Reuters. Each sample is composed of the 2000 most frequently occurring word stems in an English news story.

All datasets are preprocessed for each element before being fed into the algorithms. Precisely, we normalize all datasets to approach $\frac{1}{d}\|x_i\|_2^2$ to 1 for each $x_i \in \mathbb{R}^d$ in X .

4.2 Experimental Setting

The structure of the encoder network is set in the same way as DEC [21], SDEC [16] and FDCC [25] to be comparable with them. Concretely, we set the encoder network with dimensions of $d - 500 - 500 - 2000 - 10$ and the decoder with a symmetrical structure, where d is the dimension of input data. All layers are fully connected and activated by ReLU function except for the input, output, and embedding layers.

The parameters and centroids are initialized with a SAE and k -means in the same way as DEC [21]. Cosine similarity $\cos(a, b) = \frac{a \cdot b}{\|a\| \|b\|}$ is selected in Eq. (4, 9) for all experiments. In each iteration, we train the network with Adam optimizer. The learning rate and batch-size are set to 0.001 and 256 respectively. We investigate the influence of trade-off parameter in Eq. (5) with grid search and set it as 10. The whole training process will stop when breaks the threshold in stopping criterion $\delta = 0.001$ or reach the maximum epoch.

Table 1. Clustering performance of pairwise constraints in terms of accuracy (ACC %) and normalized mutual information (NMI %) over 5 datasets. The results of baseline models are obtained by running the released code except the ones marked by (*), which are reported from the corresponding papers. The mark (-) represents that the result is unavailable.

Dataset	MNIST		Fashion		USPS		KMNIST		Reuters10K	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
<i>k</i> -means	53.09	49.87	46.14	50.85	42.55	37.95	28.52	10.89	50.38	48.61
SAE-KM	85.23	80.76	58.03	60.57	68.75	65.99	47.16	39.10	76.53	56.61
DEC	86.59	83.73	56.62	62.21	75.81	76.91	48.64	40.79	72.17	53.08
IDEC	88.72	86.47	58.48	62.47	72.20	72.66	48.89	40.89	75.27	54.16
FCSP	62.80*	58.70*	41.70*	46.20*	-	-	-	-	-	-
COP-KM	81.60*	77.30*	54.80*	58.90*	71.85	70.24	46.78	38.53	70.42	51.83
MPC-KM	84.60*	80.80*	58.90*	61.30*	75.61	74.36	49.75	41.82	73.08	55.06
SDEC	85.02	81.69	59.62	63.89	75.84	76.96	50.05	42.18	75.31	55.24
FDCC	96.29	90.72	66.29	67.08	80.54	76.62	56.90	42.88	77.90	58.42
CDC	96.69	91.92	76.88	72.13	82.71	77.26	71.78	55.38	88.20	69.88

4.3 Experimental Results

Evaluation of Experiments on Pairwise Constraints. Our method is compared with both unsupervised clustering algorithms and constrained clustering methods. Unsupervised algorithms include *k*-means [14], *k*-means on latent feature space obtained by SAE (SAE-KM), DEC [21] and IDEC [6]. Constrained clustering algorithms include flexible CSP [19], COP-*k*means [18], MPC-*k*means [2], SDEC [16] and FDCC [25].

For the purpose of simulating human-guided constraints, we construct constraints from existing labeled data sets. We pick a set of randomly selected pairwise samples from training set and generate must-link or cannot-link constraints according to their ground truth labels. The number of constraints N is set to 3600 on MNIST, Fashion and KMNIST that accounts for merely 0.0002% of the number of possible constraints C_n^2 , and 1000 on USPS and Reuters10K that accounts for 0.0038% and 0.002% respectively. Besides, transitive constraints are also added to the known constraints. For instance, given must-link (a, b) , (a, c) and cannot-link (a, d) , we can easily deduce addable constraint: must-link (b, c) and cannot-link (b, d) , (c, d) . This conduction may cause an explosion of the constraint quantity when N is large, but can be ignored with a small amount of constraints.

The evaluation of ACC and NMI are reported in Table 1. As we can see, the performance of CDC outperforms the unsupervised algorithms with just minimal pairwise constraints. This shows that our algorithm of maintaining consistency with constraints has a positive effect on clustering. The constrained methods below are set with the same ratio of number of constraints as ours for fair comparison. The results show obvious improvement, especially on Fashion, KMNIST and Reuters10K, validating the superiority of CDC algorithm.

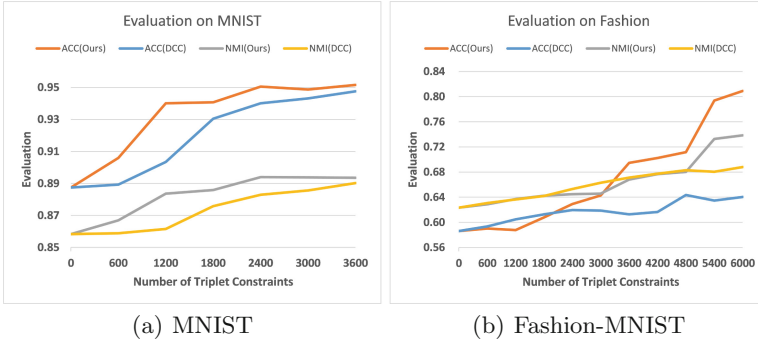


Fig. 4. Clustering ACC and NMI on MNIST and Fashion with different numbers of triplet constraints.

Evaluation of Experiments on Triplet Constraints. We evaluate the clustering performance of our method on triplet constraints by comparative experiment with FDCC [25] that put forwards triplet constraints first. To be comparable fairly with it, we introduce the same embedding space to compute Euclidean Metric among triples. Figure 4 plots the results of comparative experiment with different numbers of constraints. The results show clearly that the increase of constraint number reflects positive feedback in performance. On MNIST, minimal constraints bring about obvious improvement and then performance becomes stable, which means enough prior information has been captured. On Fashion-MNIST, the performance enhances continuously and leads to a sharp improvement in range [3000, 6000]. Comparing with FDCC, our method brings slight improvements on MNIST and obvious enhancement on Fashion-MNIST. The results validate the effectiveness of our algorithm for weak constraint information.

4.4 Parameter Analysis

We evaluate the performance with different settings of m in Eq. (9) by grid search in range [0.3, 0.6]. Figure 5 shows the parameter study results on Fashion-MNIST. Two interesting observations can be obtained: (1) The larger m produce better performance than a smaller one when given less constraints. (2) As the number of constraints increases, the results of larger m are not significantly improved or even decreased. The first observation can be explained that our objective tends to widen the difference in the similarity between positive and negative samples against the anchor, larger m enforces larger threshold to be broken down, which can promote the optimization when constraints are not enough. The second consequence occurs because our method learns enough information when more constraints are provided, a smaller m reduce the inefficiency of *imperfect triplet constraints*, which we illustrate in Sect. 3.3.

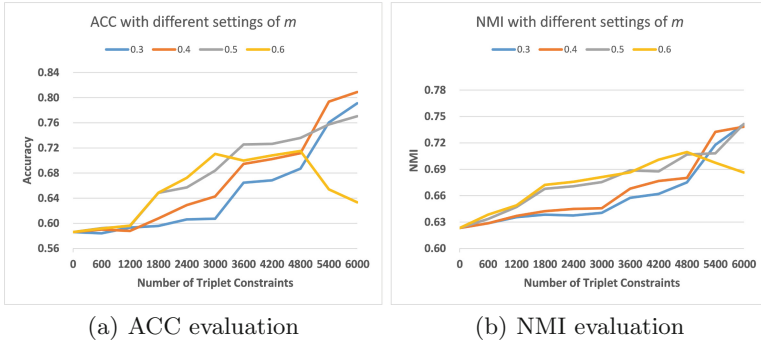


Fig. 5. The performance of our method across different settings of m on Fashion.

5 Conclusion

In this paper, we propose a Constrained Deep Clustering method (CDC) that aims to maintain consistency with constraints. The CDC method learns unsupervised information and constraint information in two consecutive modules. Effective objective functions are proposed to maintain the consistency from two perspectives simultaneously: embedding space distance and cluster assignments. Extensive experimental results on both pairwise and triplet constrained clustering validate the effectiveness of our method even if only minimal constraints are provided. Our future work will be explored from the perspective of exploring more complex similarity metrics or addressing the imbalance of the constraints.

References

1. Basu, S., Banerjee, A., Mooney, R.J.: Active semi-supervision for pairwise constrained clustering. In: Proceedings of the 2004 SIAM, pp. 333–344 (2004)
2. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: ICML, p. 11 (2004)
3. Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., Abbeel, P.: InfoGAN: interpretable representation learning by information maximizing generative adversarial nets (2016)
4. Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., Ha, D.: Deep learning for classical Japanese literature. arXiv preprint [arXiv:1812.01718](https://arxiv.org/abs/1812.01718) (2018)
5. Dilokthanakul, N., et al.: Deep unsupervised clustering with gaussian mixture variational autoencoders. arXiv preprint [arXiv:1611.02648](https://arxiv.org/abs/1611.02648) (2016)
6. Guo, X., Gao, L., Liu, X., Yin, J.: Improved deep embedded clustering with local structure preservation. In: IJCAI (2017)
7. Guo, X., et al.: Adaptive self-paced deep clustering with data augmentation. IEEE TKDE, p. 1 (2019)
8. Hsu, Y.C., Kira, Z.: Neural network-based clustering using pairwise constraints. CoRR abs/1511.06321 (2015)

9. Huang, Z., Zhou, J.T., Peng, X., Zhang, C., Lv, J.: Multi-view spectral clustering network. In: IJCAI (2019)
10. Ji, P., Zhang, T., Li, H., Salzmann, M., Reid, I.: Deep subspace clustering networks. In: NIPS (2017)
11. Jiang, Z., Zheng, Y., Tan, H., Tang, B., Zhou, H.: Variational deep embedding: an unsupervised and generative approach to clustering. arXiv preprint [arXiv:1611.05148](https://arxiv.org/abs/1611.05148) (2016)
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
13. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: Rcv1: a new benchmark collection for text categorization research. *J. Mach. Learn. Res.* **5**(4), 361–397 (2004)
14. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297. Oakland, CA, USA (1967)
15. Mukherjee, S., Asnani, H., Lin, E., Kannan, S.: Clustergan: latent space clustering in generative adversarial networks. *Proc. AAAI Conf. Artif. Intell.* **33**, 4610–4617 (2019)
16. Ren, Y., Hu, K., Dai, X., Pan, L., Hoi, S.C., Xu, Z.: Semi-supervised deep embedded clustering. *Neurocomputing* **325**, 121–130 (2019)
17. Shaham, U., Stanton, K., Li, H., Nadler, B., Basri, R., Kluger, Y.: Spectralnet: spectral clustering using deep neural networks (2018)
18. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., et al.: Constrained k-means clustering with background knowledge. *ICML* **1**, 577–584 (2001)
19. Wang, X., Davidson, I.: Flexible constrained spectral clustering. In: *SIGKDD*, pp. 563–572 (2010)
20. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint [arXiv:1708.07747](https://arxiv.org/abs/1708.07747) (2017)
21. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis (2015)
22. Yang, B., Fu, X., Sidiropoulos, N.D., Hong, M.: Towards k-means-friendly spaces: simultaneous deep learning and clustering (2017)
23. Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5147–5156 (2016)
24. Yu, Y., Zhou, W.J.: Mixture of gans for clustering. In: IJCAI (2018)
25. Zhang, H., Basu, S., Davidson, I.: A framework for deep constrained clustering - algorithms and advances. In: Brefeld, U., Fromont, E., Hotho, A., Knobbe, A., Maathuis, M., Robardet, C. (eds.) *ECML PKDD 2019. LNCS (LNAI)*, vol. 11906, pp. 57–72. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-46150-8_4
26. Zhang, T., Ji, P., Harandi, M., Huang, W., Li, H.: Neural collaborative subspace clustering (2019)
27. Zhou, L., Xiao, B., Liu, X., Zhou, J., Hancock, E.R., et al.: Latent distribution preserving deep subspace clustering. In: IJCAI. York (2019)