Latent Space Clustering via Dual Discriminator GAN

Heng-Ping He[1,2], Pei-Zhen Li[1,2], Ling Huang[1,2], Yu-Xuan Ji[1,2], and Chang-Dong Wang[1,2(✉)]

[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
hehp6@mail2.sysu.edu.cn, sysuLiPeizhen@163.com, huanglinghl@hotmail.com, jiyx6@mail2.sysu.edu.cn, changdongwang@hotmail.com
[2] Guangdong Province Key Laboratory of Computational Science, Guangzhou, China

Abstract. In recent years, deep generative models have achieved remarkable success in unsupervised learning tasks. Generative Adversarial Network (GAN) is one of the most popular generative models, which learns powerful latent representations, and hence is potential to improve clustering performance. We propose a new method termed CD2GAN for latent space **C**lustering via dual discriminator GAN (**D2GAN**) with an inverse network. In the proposed method, the continuous vector sampled from a Gaussian distribution is cascaded with the one-hot vector and then fed into the generator to better capture the categorical information. An inverse network is also introduced to map data into the separable latent space and a semi-supervised strategy is adopted to accelerate and stabilize the training process. What's more, the final clustering labels can be obtained by the cross-entropy minimization operation rather than by applying the traditional clustering methods like K-means. Extensive experiments are conducted on several real-world datasets. And the results demonstrate that our method outperforms both the GAN-based clustering methods and the traditional clustering methods.

Keywords: Latent space clustering · Unsupervised learning · D2GAN

1 Introduction

Data clustering aims at deciphering the inherent relationship of data points and obtaining reasonable clusters [1,2]. Also, as an unsupervised learning problem, data clustering has been explored through deep generative models [3–5] and the two most prominent models are Variational Autoencoder (VAE) [6] and Generative Adversarial Network (GAN) [7]. Owing to the belief that the ability to synthesize, or "create" the observed data entails some form of understanding, generative models are popular for the potential to automatically learn disentangled representations [8]. A WGAN-GP framework with an encoder for clustering is proposed in [9]. It makes the latent space of GAN feasible for clustering by carefully designing input vectors of the generator. Specifically, Mukherjee et al.

Y. Nah et al. (Eds.): DASFAA 2020, LNCS 12112, pp. 671–679, 2020.
https://doi.org/10.1007/978-3-030-59410-7_45

proposed to sample from a prior that consists of normal random variables cascaded with one-hot encoded vectors with the constraint that each mode only generates samples from a corresponding class in the original data (i.e., clusters in the latent space are separated). This insight is key to clustering in GAN's latent space. During the training process, the noise input vector of the generator is recovered by optimizing the following loss function:

$$J = ||E(G(z_n)) - z_n||_2^2 + L(E(G(z_c)), z_c) \tag{1}$$

where z_n and z_c are Gaussian and one-hot categorical components of the input vector of the generator respectively and $L(\cdot)$ denotes the cross-entropy loss function.

However, as pointed in [10], one can not take it for granted that pretty small recovery loss means pretty good features. So we propose a CD2GAN method for latent space clustering via dual discriminator GAN. The problem of mode collapse can be largely eliminated by introducing the dual discriminators. Then by focusing on recovering the one-hot vector we can obtain the final clustering result by the cross-entropy minimization operation rather than by applying the traditional clustering techniques like K-means.

2 The Proposed Method

First of all, we will give key definitions and notational conventions. In what follows, K represents the number of clusters. D_1 and D_2 denote two discriminators respectively, where D_1 favors real samples while D_2 prefers fake samples. G denotes the generator that generates fake samples and E stands for the encoder or the inverse network that serves to reduce dimensionality and get the latent code of the input sample. Let $z \in \mathbb{R}^{1\times\tilde{d}}$ denote the input of the generator, which consists of two components, i.e., $z = (z_n, z_c)$. z_n is sampled from a Gaussian distribution, i.e., $z_n \sim \mathcal{N}(\mu, \sigma^2 I_{d_n})$. In particular, we set $\mu = 0$, $\sigma = 0.1$ in our experiments and it is typical to set σ to be a small value to ensure that clusters in the latent space are separated. And $z_c \in \mathbb{R}^{1\times K}$ is a one-hot categorical vector. Specifically, $z_c = e_k, k \sim \mathcal{U}\{1, 2, \cdots, K\}$, e_k is the k-th elementary vector.

2.1 Latent Space Clustering via Dual Discriminator GAN

From Eq. (1) we know that in [9], in order to get the latent code corresponding to the real sample, it recovers the vector z_n which is sampled from a Gaussian distribution and offers no useful information for clustering. In addition, as pointed in [10], small recovery loss does not necessarily mean good features. So here comes the question, what are good features? Intuitively, they should contain the unique information of the latent vector. The above two aspects show that the latent vector corresponding to the real sample should not contain the impurity (i.e., z_n). In other words, the degree of dimension reduction is not enough. Motivated by this observation, our method only needs to perfectly recover the

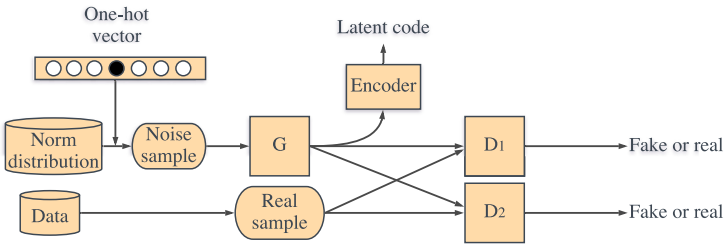

Fig. 1. The CD2GAN framework.

most unique information (i.e., z_c) of the latent vector corresponding to the real sample, and utilizes only this part for clustering.

Motivated by all above facts, we propose a dual discriminator GAN framework [11] with an inverse network architecture for latent space clustering, which is shown in Fig. 1. From [11], the Kullback-Leibler (KL) and reverse KL divergences are integrated into a unified objective function for the D2GAN framework as follows:

$$\min_{G} \max_{D1,D2} L_1(G, D_1, D_2) = \alpha E_{x\sim P_{data}}[log(D_1(x))] + E_{z\sim P_{noise}}[-D_1(G(z))] \quad (2)$$
$$+ E_{x\sim P_{data}}[-D_2(x)] + \beta E_{z\sim P_{noise}}[log(D_2(G(z))]$$

where P_{data} and P_{noise} denote the unknown real data distribution and the prior noise distribution respectively. α and β are two hyperparameters, which control the effect of KL and reverse KL divergences on the optimization problem. It is worth mentioning that the problem of mode collapse encountered in GAN can be largely eliminated since the complementary statistical properties from KL and reverse KL divergences can be exploited to effectively diversify the estimated density in capturing multi-modes.

In addition, the input noise vector (i.e., z) of the generator is z_n (sampled from a Gaussian distribution without any cluster information) concatenated with a one-hot vector $z_c \in \mathbb{R}^{1\times K}$ (K is the number of clusters). To be more precise, $z = (z_n, z_c)$. It is worth noting that each one-hot vector z_c corresponds to one type of mode provided z_n is well recovered [9]. The inverse network, i.e., the encoder, has exactly opposite structure to the generator and serves to map data into the separable latent space. The overall loss function is:

$$\min_{E,G} \max_{D1,D2} L_1(G, D_1, D_2) + L_2(E, G) + L_3(E, G) \quad (3)$$

where

$$L_2(E, G) = \lambda ||E(G(z_c)) - z_c||_2^2 \quad (4)$$

$$L_3(E, G) = \epsilon \mathcal{L}_{CE}(z_c, E(G(z_c))) \quad (5)$$

and $E(G(z_c))$ denotes the last K dimensions of $E(G(z))$, $\mathcal{L}_{CE}(\cdot)$ is the cross-entropy loss. Similar to GAN [7], the D2GAN with an inverse network model

Algorithm 1. CD2GAN

Input: Dataset X, Parameters α, β, ϵ, λ, K

for $t = 1, 2, \cdots, number\_of\_training\_epochs$ **do**
  Sample m real samples $\{x^1, ..., x^m\}$ from X
  Sample m one-hot vectors $\{z_c^1, ..., z_c^m\}$
  Sample m continuous vectors $\{z_n^1, ..., z_n^m\}$ from a Gaussian distribution ($\mu = 0, \sigma = 0.1$)
  Update discriminator D_1 by ascending along its gradient:
  $\bigtriangledown_{parameter D_1} \frac{1}{m} \sum_{i=1}^{m} [\alpha \log D_1(x^i) - D_1(G(z^i))]$
  Update discriminator D_2 by ascending along its gradient:
  $\bigtriangledown_{parameter D_2} \frac{1}{m} \sum_{i=1}^{m} [\beta \log D_2(G(z^i)) - D_2(x^i)]$
  Sample m one-hot vectors $\{z_c^1, ..., z_c^m\}$
  Sample m continuous vectors $\{z_n^1, ..., z_n^m\}$ from a Gaussian distribution ($\mu = 0, \sigma = 0.1$)
  Update generator G and the inverse network E by descending along their gradient:
  $\bigtriangledown_{parameter G,E} \frac{1}{m} \sum_{i=1}^{m} [\beta \log D_2(G(z^i)) - D_1(G(z^i)) + L_2(E, G) + L_3(E, G)]$
end for
for $i = 1, 2, \cdots, number\_of\_real\_samples$ **do**
  Get the label set Y of all x^i by **equation** (6)
end for

Output: Label set Y

can be trained by alternatively updating D_1, D_2, G and E, where Adam [12] is adopted for optimization. It is worth noting that the one-hot vector, i.e., $z_c \in \mathbb{R}^{1 \times K}$ can be regarded as the cluster label indicator of the fake sample generated from G since each one-hot vector corresponds to one mode. In addition, through adequate training, data distribution of samples generated from G will become much more identical to the real data distribution P_{data}. From this perspective, the inverse network (i.e., the encoder E) can be regarded as a muti-class classifier, which is the key reason why we introduce the cross-entropy loss $L_3(E, G)$ to the overall loss function.

When it comes to clustering, we can utilize the well trained encoder E. Formally, given a real sample x, the corresponding cluster label $\tilde{y}$ can be calculated as follows:

$$\tilde{y} = \arg\min_t \epsilon \mathcal{L}_{CE}(one\_hot(t), E(x)[\tilde{d} - K :]) + \lambda ||one\_hot(t) - E(x)[\tilde{d} - K :]||_2^2 \quad (6)$$

where $t = 0, 1, \cdots, K-1$, $one\_hot(t) \in \mathbb{R}^{1 \times K}$ is the t-th elementary vector and $E(x)[\tilde{d} - K :]$ returns a slice of $E(x)$ consisting of the last K elements. λ and ϵ remain the same as in the model training phase. For clarity, the proposed CD2GAN method is summarized in Algorithm 1.

2.2 Semi-supervised Strategy

In this section, we propose a semi-supervised strategy to accelerate and stabilize the training process of our model.

Table 1. Summary of the five datasets in the experiments.

Datasets	#Data point	Dimension	#Cluster
MNIST	70000	28×28	10
Fashion	70000	28×28	10
Pendigit	10992	1×16	10
10_x73k	73233	1×720	8
Pubmed	19717	1×500	3

From [9] we know that the one-hot component can be viewed as the label indicator of the fake samples. With the training going on, we get lots of real samples with their labels, and the encoder can be seen as a multi-class classifier. Like semi-supervised classification problem. Specifically, given a dataset with ground-truth cluster labels, a small portion (1%–2% or so) of it will be sampled along with the corresponding ground-truth cluster labels. Let $\tilde{x} = (x, y)$ denote one of the real samples for semi-supervised training, where $x \in \mathbb{R}^{1\times d}$ represents the observation while y is the corresponding cluster label. During the training process, y is encoded to a one-hot vector just like z_c, which will then be concatenated with one noise vector z_n to serve as the input of the generator G. Afterwards, the latent code of $G(z)$, i.e. $E(G(z))$, can be obtained through the encoder. Subsequently, the corresponding x will be fed into the encoder E to get another latent code $E(x)$. As mentioned above, the last K dimensions of the latent code offer the clustering information, and intuitively, one could expect better clustering performance if the the last K dimensions of $E(G(z))$ and $E(x)$ are more consistent. In light of this, a loss function can be designed accordingly:

$$L_4(E, G) = \gamma||(E(x_r) - E(G(z_n, one\_hot(x_l))))[\tilde{d} - K :]||_2^2) \tag{7}$$

where γ is a hyperparameter. In essence, we compute the Euclidean distance between the last K dimensions of $E(G(z))$ and $E(x)$, which is easy to be optimized. In this way, the overall objective function for the semi-supervised training can be defined as follows:

$$\min_{E,G} \max_{D_1,D_2} L_1(G, D_1, D_2) + L_2(E, G) + L_3(E, G) + L_4(E, G) \tag{8}$$

From the perspective of semi-supervised learning, we should sample a fixed small number of real samples when updating the parameters of E and G. In practice, faster convergence and less bad-training can be achieved when the semi-supervised training strategy is adopted.

Table 2. Comparison results for unsupervised clustering. The best result in each measure is highlighted in bold.

		MNIST	Fashion	Pendigit	10_x73k	Pubmed
NMI	ClusterGAN	0.885	0.611	0.729	0.731	0.125
	GAN with bp	0.873	0.488	0.683	0.544	0.061
	AAE	0.895	0.554	0.654	0.617	0.132
	GAN-EM	0.905	0.577	0.722	0.734	0.157
	InfoGAN	0.844	0.541	0.709	0.563	0.127
	SC	-	-	0.701	-	0.104
	AGGLO	0.677	0.565	0.681	0.599	0.112
	NMF	0.411	0.491	0.554	0.695	0.061
	CD2GAN	**0.911**	**0.638**	**0.773**	**0.783**	**0.210**
ARI	ClusterGAN	0.893	0.487	0.651	0.677	0.117
	GAN with bp	0.884	0.332	0.602	0.398	0.072
	AAE	0.906	0.425	0.590	0.546	0.112
	GAN-EM	0.902	0.399	0.642	0.659	0.132
	InfoGAN	0.825	0.398	0.632	0.401	0.102
	SC	-	-	0.598	-	0.097
	AGGLO	0.502	0.460	0.563	0.452	0.096
	NMF	0.344	0.322	0.421	0.557	0.076
	CD2GAN	**0.924**	**0.501**	**0.709**	**0.701**	**0.187**

3 Experiments

3.1 Experimental Setting

Datasets and Evaluation Measures. We adopt five well-known datasets (i.e. MNIST [13], Fashion [14], Pendigit [15], 10_x73k [9] and Pubmed [16]) in the experiments and the basic information are listed in Table 1. Two commonly used evaluation measures, i.e., normalized mutual information (NMI) and adjusted Rand index (ARI) are utilized to evaluate the clustering performance [17].

Baseline Methods and Parameter Setting. We adopt 8 clustering methods as our baseline methods including both GAN-based methods and traditional clustering methods. They are ClusterGAN [9], GAN with bp [9], adversarial autoencoder (AAE) [18], GAN-EM [19], InfoGAN [8], spectral clustering (SC) [20], Agglomerative Clustering (AGGLO) [21] and Non-negative Matrix Factorization (NMF) [22].

We set $\alpha = 0.1$, $\beta = 0.1$, $\epsilon = 1$, $\gamma = 1$ for all the datasets. As for λ, we set $\lambda = 0$ for **10_x73k** and **MNIST** and set $\lambda = 0.1$ for the other datasets. When it comes to the network architecture, we adopt some techniques as recommended in [23] for image datasets, and for the other datasets we use the full connected networks. The learning rate of **Pubmed** is set to be 0.0001 while 0.0002 for the other datasets.

Table 3. Comparison results with adopting the semi-supervised strategy for model training. The best result in each measure is highlighted in bold.

		MNIST	Fashion	Pendigit	10_x73k	Pubmed
NMI	AAE-Semi	0.943	0.721	0.810	0.823	0.299
	GAN-EM-Semi	**0.951**	0.726	0.804	0.794	0.297
	CD2GAN-Semi	0.945	**0.741**	**0.860**	**0.895**	**0.371**
ARI	AAE-Semi	0.944	0.661	0.789	0.804	0.304
	GAN-EM-Semi	**0.955**	0.653	0.754	0.788	0.311
	CD2GAN-Semi	**0.955**	**0.690**	**0.831**	**0.880**	**0.411**

3.2 Comparison Results

Comparison results are reported in Table 2 for unsupervised clustering, where the values are averaged over 5 normal training (GAN-based methods typically suffer from bad-training). As can be seen, the proposed CD2GAN method beats all the baseline methods on all the datasets in terms of the two measures. Particularly, CD2GAN is endowed with the powerful ability of representation learning, which accounts for the superiority over traditional clustering methods, i.e., SC, AGGLO and NMF. What's more, we also conduct experiments to validate the effectiveness of the semi-supervised training strategy and the results are reported in Table 3. As shown in the table, the proposed CD2GAN-Semi (CD2GAN with semi-supervised training strategy) beats AAE-Semi (AAE with semi-supervised training strategy) and GAN-EM-Semi (GAN-EM with semi-supervised training strategy) on almost all the datasets except **MNIST**.

4 Conclusion

In this paper, we propose a method termed CD2GAN for latent space clustering via D2GAN with an inverse network. Specifically, to make sure that the continuity in latent space can be preserved while different clusters in latent space can be separated, the input of the generator is carefully designed by sampling from a prior that consists of normal random variables cascaded with one-hot encoded vectors. In addition, the mode collapse problem is largely eliminated by introducing the dual discriminators and the final cluster labels can be obtained by the cross-entropy minimization operation rather than by applying traditional clustering method like K-means. What's more, a novel semi-supervised strategy is proposed to accelerate and stabilize the training process. Extensive experiments are conducted to confirm the effectiveness of the proposed methods.

Acknowledgments. This work was supported by NSFC (61876193) and Guangdong Natural Science Funds for Distinguished Young Scholar (2016A030306014).

References

1. Mai, S.T., et al.: Evolutionary active constrained clustering for obstructive sleep apnea analysis. Data Sci. Eng. **3**(4), 359–378 (2018)
2. Chen, M., Huang, L., Wang, C., Huang, D.: Multi-view spectral clustering via multi-view weighted consensus and matrix-decomposition based discretization. In: DASFAA, pp. 175–190 (2019)
3. Zhou, P., Hou, Y., Feng, J.: Deep adversarial subspace clustering. In: CVPR, pp. 1596–1604 (2018)
4. Yu, Y., Zhou, W.J.: Mixture of GANs for clustering. In: IJCAI, pp. 3047–3053 (2018)
5. Min, E., Guo, X., Liu, Q., Zhang, G., Cui, J., Long, J.: A survey of clustering with deep learning: from the perspective of network architecture. IEEE Access **6**, 39501–39514 (2018)
6. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
7. Goodfellow, I., et al.: Generative adversarial nets. In: NIPS, pp. 2672–2680 (2014)
8. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: interpretable representation learning by information maximizing generative adversarial nets. In: NIPS, pp. 2172–2180 (2016)
9. Mukherjee, S., Asnani, H., Lin, E., Kannan, S.: ClusterGAN: latent space clustering in generative adversarial networks. In: AAAI, pp. 4610–4617 (2019)
10. Hjelm, R.D., et al.: Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670 (2018)
11. Nguyen, T., Le, T., Vu, H., Phung, D.: Dual discriminator generative adversarial nets. In: NIPS, pp. 2670–2680 (2017)
12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
14. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
15. Alpaydin, E., Alimoglu, F.: Pen-based recognition of handwritten digits data set. University of California, Irvine. Mach. Learn. Repository. Irvine: University California **4**(2) (1998)
16. Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in network data. AI Mag. **29**(3), 93–106 (2008)
17. Wang, C.D., Lai, J.H., Suen, C.Y., Zhu, J.Y.: Multi-exemplar affinity propagation. IEEE Trans. Pattern Anal. Mach. Intell. **35**(9), 2223–2237 (2013)
18. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644 (2015)
19. Zhao, W., Wang, S., Xie, Z., Shi, J., Xu, C.: GAN-EM: GAN based EM learning framework. arXiv preprint arXiv:1812.00335 (2018)
20. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
21. Zhang, W., Wang, X., Zhao, D., Tang, X.: Graph degree linkage: agglomerative clustering on a directed graph. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7572, pp. 428–441. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33718-5_31

22. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788 (1999)
23. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)