



Implicit Maximum Likelihood Clustering

Georgios Vardakas and Aristidis Likas^(✉)

Department of Computer Science and Engineering, University of Ioannina,
45110 Ioannina, Greece
g.vardakas@uoi.gr, arly@cs.uoi.gr

Abstract. Clustering is a popular unsupervised machine learning and data mining problem defined as a process of assigning objects to groups so that objects in the same group are similar to each other and differ from objects in other groups. In this paper, a data clustering method is proposed that is based on unsupervised training of a generative neural network using the technique of Implicit Maximum Likelihood Estimation (IMLE). Given a dataset, IMLE is an unsupervised method that trains a neural network that takes random noise as input and produces synthetic data samples whose distribution is close to the original data. We have developed an appropriate adaptation of the IMLE generative approach that also achieves clustering of the dataset. The proposed clustering method has been evaluated on several popular datasets of various types and complexity yielding promising results.

Keywords: Clustering · Neural networks · Implicit likelihood maximization · Synthetic data generation

1 Introduction

Clustering is one type of unsupervised learning and is defined as a process of partitioning a set of objects into groups (called clusters), so that the data in the same group share common characteristics [1, 9, 20]. It is one of the most important and popular problems in machine learning and data mining with numerous applications in computer science and many other scientific and technological areas [5, 8]. Due to its particular importance, clustering is a well-studied problem and numerous approaches have been proposed that can be generally classified as hierarchical (divisive or agglomerative), model-based (e.g. k-means [14], mixture models [1]) and density-based (e.g. DBSCAN [3], DensityPeaks [18]).

A wide family of model-based approaches can be considered as Maximum Likelihood (ML) clustering methods. Such techniques construct a statistical generative model of the data by training a parametric probability density function model in order to maximize the likelihood of the data. The most popular approach is based on mixture models, where the underlying density model is a mixture of distributions. Once the mixture model has been trained, a clustering solution can be directly obtained by assuming that each component distribution

corresponds to a cluster and computing the posterior probability $P(k|x)$ that each data object x has been generated from the k -th component distribution. Then each data point x is assigned to the cluster k with maximum posterior probability. In the case where the component distributions are Gaussian, the well-known Gaussian mixture model (GMM) is obtained. An important issue to be stressed is that training neural network models for maximum likelihood clustering is considered a difficult task [16].

In this work, we aim to achieve generative maximum likelihood clustering based on neural networks, by exploiting a recently proposed method of *Implicit Maximum Likelihood Estimation* (IMLE) [13]. Given a set of data objects X , this method uses a neural network (called generator) that takes random input vectors and produces synthetic samples in the data space. By minimizing an appropriate objective, the network is trained so the distribution of samples resembles the data distribution. A notable issue is that it is proved that this training procedure maximizes the likelihood of the dataset without explicitly computing the likelihood.

Our proposed clustering method appropriately adapts the IMLE approach in order to achieve maximum likelihood clustering based on neural networks. As it will be explained, the modification occurs both in the way that the random input vectors are generated and in the way that representative synthetic samples are selected in order to be used for training. The method finally provides both a neural generator of synthetic samples that resemble the objects of dataset X as well as a partitioning of X into clusters.

The organization of the paper is the following. In Sect. 2 the IMLE method is described, while in Sect. 3 the proposed IMLE clustering method is presented and explained. Section 4 presents comparative experimental results on various datasets, while Sect. 5 provides conclusions and directions for future research.

2 Generative Modeling Using IMLE

Suppose we are given a dataset $X = \{x_1, \dots, x_N\}$ where $x_i \in \mathbb{R}^d$. The Implicit Maximum Likelihood Estimation (IMLE) [13] approach assumes a neural network \mathcal{G}_θ with m inputs, d outputs and parameter vector (weights) θ . This network (called generator) takes as input a random vector $z \in \mathbb{R}^m$ usually sampled from the Normal distribution and produces a sample $s^\theta \in \mathbb{R}^d$, i.e., $s^\theta = \mathcal{G}_\theta(z)$ (see Fig. 2a).

IMLE trains the generator so that it can generate synthetic samples s^θ that resemble the real data x . It is a simple generative method that under certain conditions is equivalent to maximum likelihood estimation. This is surprising given that the IMLE objective does not explicitly contain any log-likelihood term and training neural networks using maximum likelihood is considered a difficult task [16].

At each IMLE iteration a sampling procedure takes place where a set of L random input vectors (called latent variables) are drawn from the Normal distribution $z_i \sim \mathcal{N}(0, \sigma^2)$ and used for the computation of the synthetic samples

$s_i^\theta = \mathcal{G}_\theta(z_i)$ ($i = 1, \dots, L$). Then, for each real data example x_i ($i = 1, \dots, N$), its representative sample $r_i^\theta \in S^\theta$ is computed via an application of the nearest neighbor search (NNS) in S^θ based on Euclidean distance, i.e. $r_i^\theta = NNS(x_i, S^\theta)$. The generator parameters θ are updated in order to minimize the following IMLE objective function:

$$\hat{\theta}_{IMLE} = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^n \|r_i^\theta - x_i\|_2^2 \tag{1}$$

IMLE training is summarized in Fig. 1. It is obvious that IMLE is very simple to implement. Moreover, it does not suffer from mode collapse, vanishing gradients or training instability, unlike popular deep generative methods such as, for example, GANs [7]. Mode collapses do not occur since the loss ensures that each data example is represented by at least one sample. Gradients do not vanish because the gradient of the distance between a data example and its representative sample does not become zero unless they coincide. Training is stable because the IMLE estimator is the solution to a simple minimization problem. Finally, it can be used both in the case of small and large datasets.

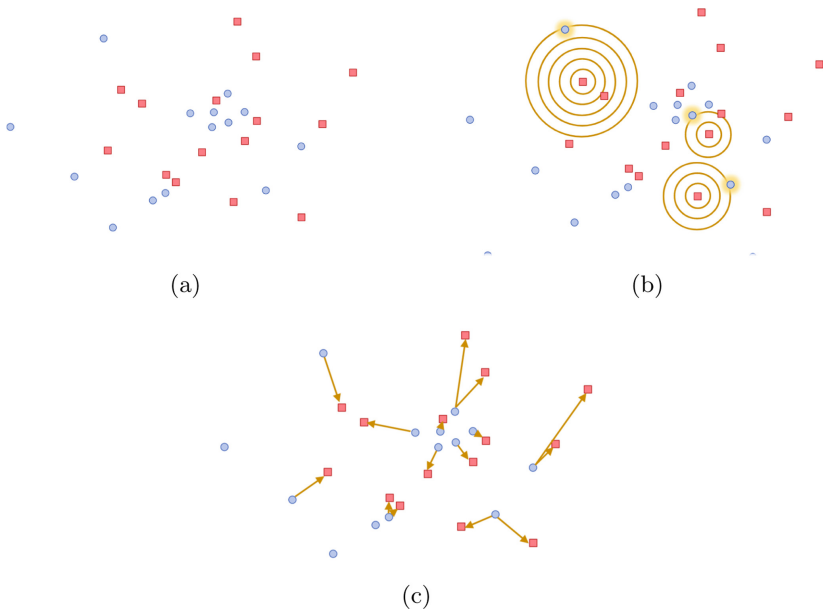


Fig. 1. (a) The data points are represented by squares and the samples by circles. (b) For each data point the nearest sample is found. (c) Minimize the IMLE objective.

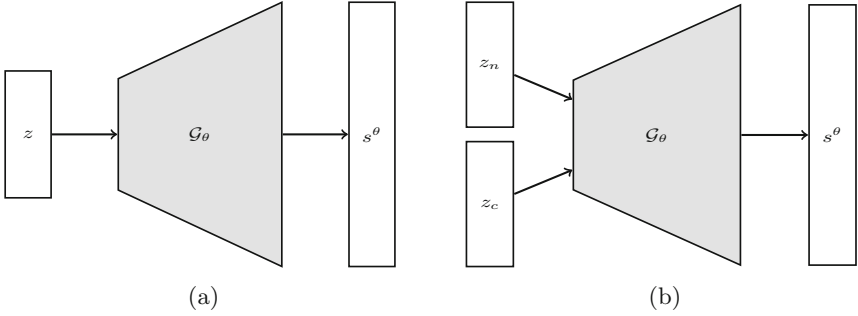


Fig. 2. (a) IMLE general architecture. (b) IMLE clustering architecture.

3 Clustering Based on IMLE

In this work, we propose a modification of the IMLE method in order to achieve not only synthetic data generation but also clustering of the original dataset X . More precisely, we introduce two basic algorithmic modifications to the IMLE framework namely: (i) a cluster-friendly sampling prior to generate random input vectors z_i and (ii) the two-stage nearest neighbor search algorithm to determine the representative sample $r_i^\theta \in S^\theta$ for a data point x_i . The two issues are explained below.

An important observation is that each data point x can be associated with the input vector z that generated the representative sample r^θ of x . Therefore, except for the correspondence (x, r^θ) , the correspondence (x, z) (where $r^\theta = G_\theta(z)$) can be defined and it will be exploited in our clustering method.

3.1 Cluster Friendly Input Distribution

In the original IMLE method, the input random vectors z belong to a single cluster since they are drawn from a multivariate m -dimensional Normal distribution. However, it can be observed that, if the original data x_i form distinct clusters, the corresponding input vectors also demonstrate a clustering tendency in the sense that similar data points x_i correspond to input vectors z_i that are close.

Based on this observation, if we assume that the random input vectors are drawn from a mixture model (i.e. from K distinct distributions), then a clustering of the original dataset X can be obtained: each data point x_i can be assigned to the cluster from which its corresponding input vector has been drawn. Therefore in the proposed method, the single Normal distribution is replaced by K non-overlapping distributions, with the k -th distribution responsible for the generation of the subset Z_k of input vectors assigned to cluster k . The most obvious first choice is a mixture of K m -dimensional Gaussian distributions. However, this choice requires the specification of the centers and covariances of K Gaussian distributions which are well separated.

A more sophisticated mechanism for generating m-dimensional random vectors that form K disjoint clusters has been proposed in [17], where input vector z consists of two parts, i.e. $z = (z_c, z_n)$. The first deterministic part z_c which is the one-hot encoding of the corresponding cluster, thus for K clusters the dimension of z_c is equal to K . The second part z_n is randomly drawn a p-dimensional Gaussian distribution: $z_n \sim \mathcal{N}(0, \sigma^2 I)$. In addition, the standard deviation σ is set to a small value like $\sigma = 0.10$ to ensure that the K clusters of random vectors Z_k do not overlap.

In summary, in order to generate an input vector $z = (z_c, z_n)$ of cluster k , we set z_c equal to the one-hot encoding of k and draw z_n from $\mathcal{N}(0, \sigma^2 I)$. By sampling an equal number of vectors for each cluster k the set of random input vectors Z is created at each iteration which is partitioned into disjoint subsets Z_k each one containing the random input vectors for cluster k ($k = 1, \dots, K$). Additionally, since $s^\theta = G_\theta(z)$, the set S^θ of computed samples is partitioned into K disjoint clusters S_k^θ . Consequently, the original dataset X can be partitioned into K clusters by assigning each x_i to the cluster of its representative r_i^θ , i.e. if $r_i^\theta \in S_k^\theta$ then x_i is assigned to cluster k .

3.2 Two-Stage Nearest Neighbor Search for Determining Data Representatives

As we already mentioned, for each data point x_i , the IMLE method performs a nearest neighbor search in the entire set of the generated samples S^θ to locate the representative sample $r_i^\theta \in S^\theta$ that is used in the IMLE objective (Eq. 1). We have empirically observed that clustering performance can be improved if we modify this strategy in order to take into account that samples s^θ are partitioned into subsets S_k^θ .

To take this information into account, we first compute the centroid c_k of each subset S_k^θ . Then we assign a data point x_i to the cluster l whose centroid c_l is nearest to x_i based on Euclidean distance. Then instead of determining the representative sample for x_i through nearest neighbor search over the entire set of samples S^θ , we confine the nearest neighbor search to the specific subset S_l^θ that contains the samples of cluster l . Therefore, $\hat{r}_i^\theta = NNS(x_i, S_l^\theta)$.

$$\hat{\theta} := \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^n \|\hat{r}_i^\theta - x_i\|_2^2 \tag{2}$$

The proposed IMLE clustering method is presented in the Algorithm 1. Note that the gradient-based updates can use any standard gradient-based learning rule with the most popular choice being the Adam optimizer [10].

Algorithm 1. IMLE clustering algorithm**Require:** Data X , number of clusters K

- 1: Specify the network architecture \mathcal{G}_θ , the number of samples L , the number of iterations T
- 2: **for** $t \leftarrow 1$ to T **do**
- 3: Sample K clusters of random input vectors $Z = \{z_1, \dots, z_L\}$, where $z = (z_n, z_c)$. Let Z_k the subset corresponding to cluster k .
- 4: Generate samples $\{s_1^\theta, \dots, s_L^\theta\}$, where $s_i^\theta = \mathcal{G}_\theta(z_i)$. Let S_k^θ the subset corresponding to cluster k .
- 5: For every S_k^θ compute the corresponding centroid c_k .
- 6: For each x_i find \hat{r}_i^θ with the two-stage nearest neighbor search.
- 7: Update the parameters θ of the generator network by descending its stochastic gradient:

$$\nabla_\theta \sum_{i=1}^n \|\hat{r}_i^\theta - x_i\|_2^2 \quad (3)$$

8: **end for**9: Find the final clustering by assigning each x_i based on the nearest sample centroid c_l .10: Return the final network parameters θ^* and the clustering solution.**Table 1.** Main characteristics of the tested datasets.

Dataset	Instances	Features	Classes
<i>Images</i>			
Fashion-Mnist	60000	784	10
Mnist	60000	784	10
Olivetti	400	4096	40
<i>Tabular</i>			
10x_73k	73233	720	8
Australian	690	14	2
Dermatology	366	34	6
Ecoli	336	7	8
Iris	150	4	3
Pendigits	10992	16	10
Wine	178	13	3

4 Experiments

4.1 Datasets

In order to evaluate the proposed clustering method, several datasets have been considered. More specifically, we have conducted experiments on three image datasets and seven tabular datasets. In Table 1, we present the characteristics

of the datasets used in our study. In all experiments, the number of clusters K was set equal to the number of classes of each dataset. All data are normalized in $[0, 1]$.

- **Fashion-MNIST** [19] dataset consists 60,000 training samples of grayscale images describing a fashion product. Each sample is a 784 image associated with a label from ten classes.
- **MNIST** [12] dataset consists 60,000 training samples of grayscale images of handwritten digits. Each sample is a 784 image and it is associated with a label from ten classes.
- **Olivetti** [6] is a face database of 40 individuals with ten 4096 grayscale images per individual. For some individuals, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement).
- **10x_73k** [21] dataset consists of 73233 RNA-transcripts belonging to 8 different cell types. The 720 genes with the highest variances across the cells were selected to reduce the data dimensionality. The data set is sparse, since the data matrix has about 40% zero values.
- **Australian** [2] two-class dataset is composed of 690 credit card applications. Each sample is described by 14-dimensional feature vector.
- **Dermatology** [2] six-class dataset is composed of 366 patient records that suffer from six different types of the Eryhemato-Squamous disease. Each patient is described by a 34-dimensional vector, containing clinical and histopathological features.
- **Ecoli** [2] includes 336 proteins from the E.coli bacterium and seven attributes, calculated from the amino acid sequences, are provided. Proteins belong to eight categories according to their cellular localization sites.
- **Iris** [2] dataset contains three classes of 50 instances each, where each class refers to a type of iris plant. Each sample is described by a 4-dimensional vector, corresponding to the length and the width of the sepals and petals, in centimeters.
- **Pendigits** [2] dataset consists of 10992 writing samples from 44 different writers, in total 10992 written samples. Each sample is a 16-dimensional vector, containing pixel coordinates associated with a label from ten classes.
- **Wine** [2] tree-class dataset consists of 178 samples of chemical analysis of wines. Each sample is described by a 13-dimensional feature vector.

4.2 Neural Network Architecture

The networks are trained using the Adam optimizer with learning rate $n = 10^{-4}$ and coefficients $b_1 = 0.5$ and $b_2 = 0.9$. The dimension of z_c is the set equal to the number of classes in the dataset. We used Leaky Relu activations (LRelu) with leak = 0.2 and Batch Normalization (BN) and we trained the generator for 2000

Table 2. Network architecture for each dataset.

Dataset	Input (z_n, z_c)	Hidden {1, 2}	Output
<i>Images</i>			
Fashion-MNIST	(10, 10)	FC 256 LReLU BN	FC 784 Sigmoid
MNIST	(10, 10)	FC 256 LReLU BN	FC 784 Sigmoid
Olivetti	(10, 40)	FC 256 LReLU BN	FC 4096 Sigmoid
<i>Tabular</i>			
10x_73k	(10, 8)	FC 256 LReLU BN	FC 720 Sigmoid
Australian	(5, 2)	FC 256 LReLU BN	FC 14 Sigmoid
Dermatology	(5, 6)	FC 256 LReLU BN	FC 34 Sigmoid
Ecoli	(2, 8)	FC 256 LReLU BN	FC 7 Sigmoid
Iris	(2, 3)	FC 256 LReLU BN	FC 4 Sigmoid
Pendigits	(5, 10)	FC 256 LReLU BN	FC 16 Sigmoid
Wine	(2, 3)	FC 256 LReLU BN	FC 3 Sigmoid

Table 3. Selected batch size per datasets.

Dataset	Instances	Batch size	#Batches
<i>Images</i>			
Fashion-Mnist	60000	1024	58
Mnist	60000	1024	58
Olivetti	400	400	1
<i>Tabular</i>			
10x_73k	73233	1024	71
Australian	690	690	1
Dermatology	366	366	1
Ecoli	336	336	1
Iris	150	150	1
Pendigits	10992	1024	10
Wine	178	178	1

epochs for all datasets. The hidden layers are the same for all networks. It is necessary to adjust the input and the output layers based on the given dataset. The details of the network architectures are presented in Table 2. The number of synthetic samples generated was chosen to be equal to twice the number of data ($L = 2N$) in all cases. In Table 3 we present the selected batch size per dataset.

4.3 Evaluation Metrics

It is necessary to mention that since clustering is an unsupervised problem, we ensured that all algorithms are unaware of the true category of the data. In order to evaluate the results of the clustering methods, we use the standard evaluation metrics which assume that a ground truth clustering is available. For all algorithms, the number of clusters is set to the number of ground-truth categories [15] and assumes ground truth that cluster labels coincide with class labels. The first evaluation metric is Clustering Accuracy (ACC):

$$ACC = \max_m \frac{\sum_{i=1}^n I(y_i = m(c_i))}{n} \quad (4)$$

where $I(x) = 1$ if x is true and 0 otherwise, y_i is the ground-truth label, c_i is the cluster assignment generated by the clustering algorithm, and m is a mapping function which ranges over all possible one-to-one mappings between assignments and labels. This metric finds the best matching between cluster assignments from a clustering method and the ground truth. It is worth noting that the optimal mapping function can be efficiently computed by the Hungarian algorithm [11]. The second evaluation metric is the Normalized Mutual Information (NMI) defined as [4]:

$$NMI(Y, C) = \frac{2 \times I(Y, C)}{H(Y) + H(C)} \quad (5)$$

where Y denotes the ground-truth labels, C denotes the clusters labels, I is the mutual information metric and H the entropy.

4.4 Experimental Results

In our experimental study, the proposed IMLE clustering method was compared in all datasets against k-means and the typical maximum likelihood clustering method which is the Gaussian Mixture Model (GMM). It should be noted that GMMs with diagonal covariance has been considered. Since all compared methods depend on initialization, we executed each algorithm 10 times with random initialization and provide in Table 4 the average and standard deviation for ACC and NMI.

It can be observed that the IMLE clustering approach outperforms the typical methods in the case of large datasets with structured data (images) in most cases. For small datasets, it is superior in some cases, while in the remaining cases it demonstrates comparable performance. It should be emphasized that the method does not necessarily require large datasets to be trained as happens with deep clustering methods (like clusterGAN [17]) that cannot be employed to cluster datasets with few data. This major advantage of our method is inherited from the IMLE approach and makes the method applicable in all clustering problems.

Table 4. Experimental results on several datasets. Bold numbers indicate the best average performance for each dataset.

Dataset	Algorithm	ACC	NMI
<i>Images</i>			
Fashion-MNIST	IMLE clustering	0.56 ± 0.05	0.51 ± 0.02
	K-means	0.52 ± 0.02	0.47 ± 0.01
	GMM	0.52 ± 0.02	0.48 ± 0.02
MNIST	IMLE clustering	0.55 ± 0.02	0.48 ± 0.02
	K-means	0.51 ± 0.03	0.48 ± 0.02
	GMM	0.51 ± 0.03	0.49 ± 0.02
Olivetti-Faces	IMLE clustering	0.56 ± 0.03	0.77 ± 0.01
	K-means	0.52 ± 0.02	0.74 ± 0.01
	GMM	0.35 ± 0.03	0.62 ± 0.03
<i>Tabular</i>			
10x_73k	IMLE clustering	0.53 ± 0.03	0.49 ± 0.03
	K-means	0.54 ± 0.05	0.55 ± 0.04
	GMM	0.54 ± 0.04	0.58 ± 0.01
Australian	IMLE clustering	0.77 ± 0.07	0.25 ± 0.14
	K-means	0.73 ± 0.13	0.23 ± 0.18
	GMM	0.67 ± 0.11	0.12 ± 0.12
Dermatology	IMLE clustering	0.73 ± 0.05	0.80 ± 0.04
	K-means	0.68 ± 0.13	0.80 ± 0.06
	GMM	0.70 ± 0.04	0.70 ± 0.04
Ecoli	IMLE clustering	0.56 ± 0.03	0.56 ± 0.03
	K-means	0.52 ± 0.05	0.56 ± 0.03
	GMM	0.58 ± 0.08	0.60 ± 0.03
Iris	IMLE clustering	0.89 ± 0.01	0.74 ± 0.02
	K-means	0.82 ± 0.10	0.70 ± 0.05
	GMM	0.92 ± 0.00	0.80 ± 0.00
Pendigits	IMLE clustering	0.70 ± 0.03	0.66 ± 0.03
	K-means	0.69 ± 0.04	0.69 ± 0.01
	GMM	0.58 ± 0.03	0.58 ± 0.03
Wine	IMLE clustering	0.94 ± 0.01	0.82 ± 0.02
	K-means	0.92 ± 0.08	0.80 ± 0.08
	GMM	0.94 ± 0.03	0.82 ± 0.06

5 Conclusions

We have proposed a data clustering method that is based on training a generative neural network using the technique of Implicit Maximum Likelihood Estimation (IMLE). In IMLE a neural network is trained that takes as input random noise and produces synthetic data similar to the data in the training set. We have appropriately modified the IMLE method by combining the generative process with a clustering procedure in order to perform clustering of the data in the training set. The proposed method has provided good clustering results on several datasets of various sizes and dimensionality.

Future research could focus on the detailed experimental investigation of the performance of the method and its sensitivity to various parameters such as the network architecture and the number of synthetic samples. Alternative mixture distributions for the random inputs could also be examined. Finally, it would be interesting to consider the use of a second neural network that will be trained to implement the inverse mapping of the generator network, i.e. it will take a synthetic sample as input and will provide as output the corresponding random input vector.

Acknowledgements. This research was supported by project “Dioni: Computing Infrastructure for Big-Data Processing and Analysis” (MIS No. 5047222) co-funded by European Union (ERDF) and Greece through Operational Program “Competitiveness, Entrepreneurship and Innovation”, NSRF 2014-2020.

References

1. Bishop, C.M.: Pattern recognition and machine learning, Springer (2006)
2. Dua, D., Graff, C.: UCI machine learning repository (2017). <http://archive.ics.uci.edu/ml>
3. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol. 96, pp. 226–231 (1996)
4. Estévez, P.A., Tesmer, M., Perez, C.A., Zurada, J.M.: Normalized mutual information feature selection. *IEEE Trans. Neural Netw.* **20**(2), 189–201 (2009)
5. Filippone, M., Camastra, F., Masulli, F., Rovetta, S.: A survey of kernel and spectral methods for clustering. *Pattern Recogn.* **41**(1), 176–190 (2008)
6. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. *Science* **315**(5814), 972–976 (2007)
7. Goodfellow, I., et al.: Generative adversarial nets. In: *Advances in Neural Information Processing Systems*, pp. 2672–2680 (2014)
8. Jain, A.K.: Data clustering: 50 years beyond k-means. *Pattern Recogn. Lett.* **31**(8), 651–666 (2010)
9. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM Comput. Surv. (CSUR)* **31**(3), 264–323 (1999)
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)* (2014)
11. Kuhn, H.W.: The Hungarian method for the assignment problem. *Nav. Res. Logist. (NRL)* **52**(1), 7–21 (2005)

12. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010). <http://yann.lecun.com/exdb/mnist/>
13. Li, K., Malik, J.: Implicit maximum likelihood estimation. arXiv preprint [arXiv:1809.09087](https://arxiv.org/abs/1809.09087) (2018)
14. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, vol. 1, pp. 281–297 (1967)
15. 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)
16. Mohamed, S., Lakshminarayanan, B.: Learning in implicit generative models. arXiv preprint [arXiv:1610.03483](https://arxiv.org/abs/1610.03483) (2016)
17. Mukherjee, S., Asnani, H., Lin, E., Kannan, S.: ClusterGAN: latent space clustering in generative adversarial networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 4610–4617 (2019)
18. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. *Science* **344**(6191), 1492–1496 (2014)
19. 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)
20. Xu, R., Wunsch, D.: Clustering, vol. 10. Wiley (2008)
21. Zheng, G.X., et al.: Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**(1), 1–12 (2017)