High-dimensional Data Clustering with Fuzzy C-Means: Problem, Reason, and Solution*

Yinghua Shen[1], Hanyu E[2], Tianhua Chen[3], Zhi Xiao[1], Bingsheng Liu[4], Yuan Chen[5]
[1] School of Economics and Business Administration, Chongqing University, China
{yinghua, xiaozhi}@cqu.edu.cn
[2] Department of Electrical and Computer Engineering, University of Alberta, Canada
hanyu6@ualberta.ca
[3] Department of Computer Science, University of Huddersfield, UK
t.chen@hud.ac.uk
[4] School of Public Affairs, Chongqing University, China
bluesea_boy_1979@163.com
[5] College of Management and Economics, Tianjin University, China
yuanchen_2020@tju.edu.cn

Abstract. Fuzzy C-Means (FCM) clustering algorithm is a popular unsupervised learning approach that has been extensively utilized in various domains. However, in this study, we point out a major problem faced by FCM when it is applied to the high-dimensional data, i.e., quite often the obtained prototypes (cluster centers) could not be distinguished with each other. Many studies have claimed that the concentration of the distance (CoD) could be a major reason for this phenomenon. This paper has therefore revisited this factor, and highlight that the CoD could not only lead to decreased performance, but sometimes also positively contribute to enhanced performance of the clustering algorithm. Instead, this paper point out the significance of features that are noisy and correlated, which could have a negative effect on FCM performance. Hence, to tackle the mentioned problem, we resort to a neural network model, i.e., the autoencoder, to reduce the dimensionality of the feature space while extracting features that are most informative. We conduct several experiments to show the validity of the proposed strategy for the FCM algorithm.
Keywords: Fuzzy C-Means; High-dimensional data; Autoencoder.

1. Introductory Note

Clustering is one of the most important techniques used to explore the structure of data. It intends to gather those data points close (in terms of distance, similarity, functionality, etc.) to each other into a group and distributes those far apart from each other into the different groups. Many different

* This work was supported in part by the National Natural Science Foundation of China under Grant 72001032, Grant 72071021, Grant 72002152; in part by Natural Science Foundation of Chongqing under Grant cstc2020jcyj-bshX0013.

kinds of clustering concepts and algorithms have been proposed so far, which could be roughly classified into partition-based methods [1]–[3], graph-based methods [4], [5], hierarchy-based methods [6], [7], and density-based methods [8], [9]. Among these methods, the fuzzy partition-based methods, e.g., Fuzzy C-Means (FCM) [2], [3], [10]–[12], which bring the concept of fuzzy set [13] into clustering, have seen a rapid development in both theory and real-world applications. By assigning the cluster membership degree, which is a value in the interval [0, 1], to a certain data point, the structure of data could be described by some overlapped clusters which are more suitable to represent and handle the complex phenomena in real world. We briefly review the concept and algorithm of the FCM as follows.

Suppose that we have the data set as $X = (\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_N)^{\mathrm{T}}$, $\boldsymbol{x}_k$ is the k-th data point in the n dimensional feature space $\boldsymbol{R}^n$. The generic version of the FCM algorithm [3] minimizes the following objective function with the (weighted) Euclidean distance as

$$Q = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m} \left\| \boldsymbol{x}_k - \boldsymbol{v}_i \right\|^2 \tag{1}$$

with the distance expressed as

$$\left\| \boldsymbol{x}_k - \boldsymbol{v}_i \right\|^2 = \sum_{j=1}^{n} \frac{(x_{kj} - v_{ij})^2}{\sigma_j^2} \tag{2}$$

where σ_j is a standard deviation of the j-th variable of the data, and fuzzification coefficient m is usually greater than 1. The data is partitioned into c clusters coming in the form of the partition matrix $U = [u_{ik}]_{c\times N}$, $i = 1, 2,\ldots, c$; $k = 1, 2,.., N$, with a collection of prototypes represented as $V = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c)^{\mathrm{T}}$. The k-th data is described in terms of the k-th column membership grades in the partition matrix. By the alternating optimization (AO) algorithm in [14], each element in the partition matrix is calculated as

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\dfrac{\left\| \boldsymbol{x}_k - \boldsymbol{v}_i \right\|}{\left\| \boldsymbol{x}_k - \boldsymbol{v}_j \right\|} \right)^{2/(m-1)}} \tag{3}$$

and each entry in the prototype is obtained as

$$v_{it} = \frac{\sum_{k=1}^{N} u_{ik}^{m} x_{kt}}{\sum_{k=1}^{N} u_{ik}^{m}} \tag{4}$$

where $t = 1, 2,\ldots, n$.

However, one problem with FCM is that it may not work well when the dimensionality n of the feature space is high, because the prototypes found by the algorithm could be quite similar to each other. To illustrate this phenomenon, we use two high-dimensional data sets from the UCI machine learning data repository, i.e., Isolet (with 1560 samples and 617 features) and Hand (with 1800 samples and 3000 features). We apply the FCM to these two data sets with both the cluster number and fuzzification coefficient set to 2. We observe that for each data set, the obtained two prototypes are exactly the same. Due to space limit, we only select the clustering results of the first 10 features, as documented in Table 1. Clearly, these are not desired results, thus motivating this

research to resort to the autoencoder to mitigate this issue when applying FCM in high-dimensional feature space.

In the following, we first analyze why sometimes the FCM does not work well in the high-dimensional feature space in Section 2. We use the autoencoder, a neural network model, to reduce the feature space in Section 3. In Section 4, we conduct several experimental studies to demonstrate the validity of the propose strategy to make FCM work in the high-dimensional feature space. Finally, we conclude the paper and point out some future studies in Section 5.

Table 1 Results of the first 10 features of the obtained prototypes.

Data Sets	Prototypes	Features									
		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
Isolet	$\boldsymbol{v}_1$	0.35	0.55	0.67	0.72	0.68	0.63	0.60	0.54	0.49	0.46
	$\boldsymbol{v}_2$	0.35	0.55	0.67	0.72	0.68	0.63	0.60	0.54	0.49	0.46
Hand	$\boldsymbol{v}_1$	0.30	0.40	0.53	0.71	0.28	0.50	0.49	0.70	0.45	0.44
	$\boldsymbol{v}_2$	0.30	0.40	0.53	0.71	0.28	0.50	0.49	0.70	0.45	0.44

2. Reasons for Failure of FCM

Concentration of distance (CoD) has been discovered as one of the major aspects of the curse of dimensionality [15]–[17]. The general statement for this phenomenon is that, under certain assumptions such as data points are obtained from the independent and identical distributions, data points will become close to each other making them indistinguishable. Hence, a follow-up question is that will this concentration seriously affect the algorithms which are based on the distance measure? In spite of evidences that CoD may have made the K-nearest neighbor (KNN) unstable [15], [18], [19], research has been undergoing to identify its comprehensive effects on classification and clustering algorithms. Specifically, [20] did find that the CoD can be used to improve the classification accuracy of the algorithm. For the clustering algorithm, [21] observed that CoD does not always have a negative effect on clustering. In case each feature of the data sets contributes to the clusters contained therein, CoD is helpful in distinguishing the clusters; however, when the generated clusters mainly result from a small number of features, with the remaining being noisy features (e.g., those satisfying the normal distribution), CoD can make the clusters merged together.

It seems that, performance of the clustering or classification algorithm does not totally depend on the CoD. It is determined by the relationship between the embedding dimension and intrinsic dimension. In fact, when we have a real-world high-dimensional data, the concentration degree is not necessarily high with Table 1 in [16] as an example. It has been pointed out that high-dimensional (in terms of the embedding dimension) real-world data usually has a much lower intrinsic dimensionality, which is a consensus in the high-dimensional data analysis community [22]. In fact, many approaches have been proposed to estimate this intrinsic dimension, from columns d and d_{mle} in Table 1 in [23] we can catch a glimpse of the relationship between the embedding and intrinsic dimensions. As will be shown in our experiment, the size of intrinsic

dimensions does not directly contribute to the occurrence of the CoD. Also, we will see that CoD could be beneficial or detrimental to clustering, but high intrinsic dimension is beneficial for clustering while high embedding dimension with low intrinsic dimension is not good for clustering.

To illustrate this, in the following we design and implement 3 experiments (with generated synthetic data) which corresponds to 3 scenarios. We want to check the separability of the clusters in a data set in relation to the increasing dimensionality. Scenario 1: each feature has a multimodal distribution (a mixture of two Gaussian distributions), features are independent with each other. Scenario 2: the first feature has a multimodal distribution, while other features have the same Gaussian distributions. Scenario 3: all the features are linearly related, and each feature has a multimodal distribution. The design of the experiments are inspired by that in [21], where Scenarios 1 and 2 (the histogram parts) are repeated for two examples in [21].

Scenario 1: Suppose we have a data set X with $N = 2000$ observations and n features. X consists two equal-sized clusters satisfying the multivariable Gaussian distribution, with the cluster centers as $\text{center1} = [0, 0, ..., 0]_{1\times n}$ and $\text{center2} = [1, 1, ..., 1]_{1\times n}$, with the covariance matrix (to model the spread of each cluster) derived by multiplying an n-dimensional identity matrix by the constant *variance* = 1. An example of such data set when $n = 2$ is given in Fig.1. This setting makes the formed data set X have the same size of intrinsic and embedding dimensions because each newly formed feature provides the information of two clusters. We measure the Euclidean distance between each point in X and the origin of the n-dimensional space (i.e., the Euclidean norm), then give the histogram of these distances. We illustrate the results in Fig. 2 when n is set as 1, 100, and 1000, respectively.

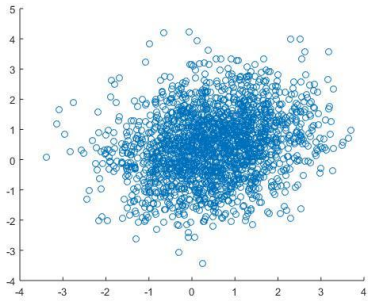


Fig. 1. Data set X when $n = 2$.

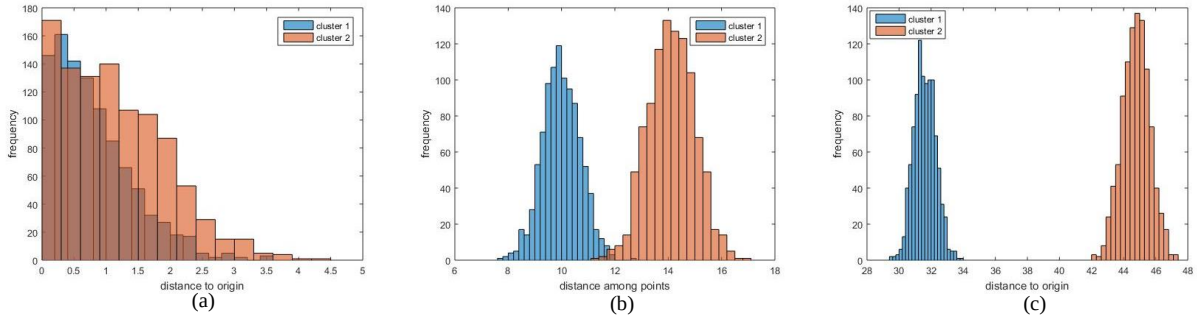


Fig. 2. Histogram of the distances to the origin when (a) $n = 1$; (b) $n = 100$; and (c) $n = 1000$.

Obviously, when n is small, distances from these two clusters to the origin have a similar distribution, and large portion of the distributions are overlapped with each other. However, this overlap becomes increasingly smaller as the dimensionality increases. When n is 1000, these two distributions are completely separated with each other, which means that the two clusters are well separated. In fact, we could also consider the distances among each cluster (intra cluster distances) and those among the clusters (inter cluster distances). We show the histogram of these distances with the increasing feature dimensionality in Fig. 3 when n is 1, 100, and 1000, respectively. Obviously, with the increasing dimensionality, intra cluster distances or inter cluster distances are increasing, but latter increase more rapidly than the former.

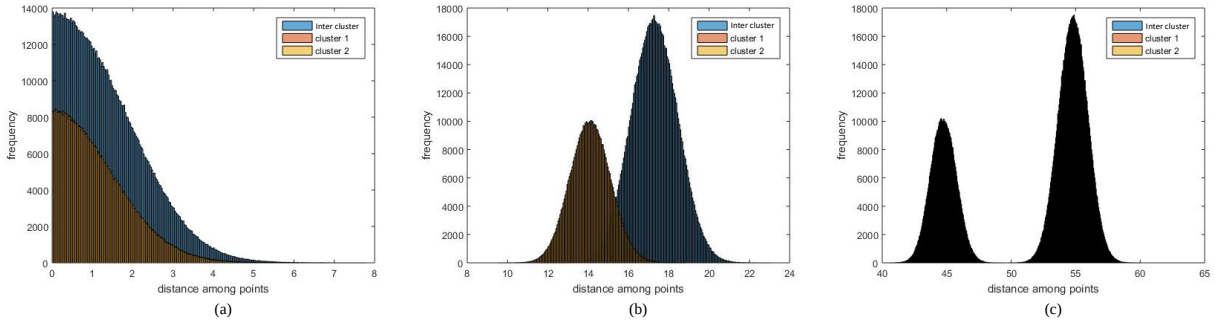


Fig. 3. Histogram of the inter cluster and intra cluster distances when (a) n = 1; (b) n = 100; and (c) n = 1000.

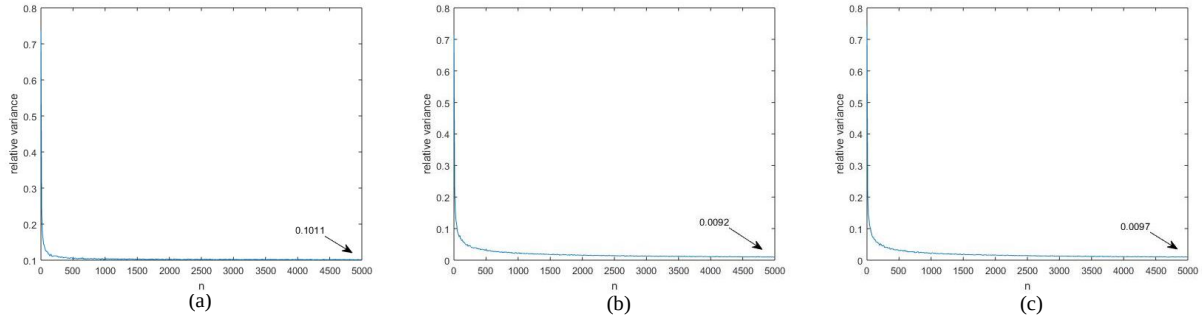


Fig. 4. Concentration degree for the whole data set and two clusters.

In the following, we intend to check the CoD phenomenon in this data set. The sufficient and necessary condition to incur the CoD was raised in [18] and [24], that is, the relative variance has to converge to 0 (or the relative contrast has to converge to 0 in probability 1) when feature dimensionality increases to infinite. Here, the relative variance is derived by diving the standard deviation of the distances among data points by the expectation of these distances. Experimentally we could show that in this scenario, when we range the feature dimensionality n from 1 to 5000 with a step size of 10, the relative variance will converge to a positive constant around 0.1 in Fig. 4(a). However, as for the two clusters, this relative variance reaches the value around 0.01 in Figs. 4(b) and (c), and obviously relative variance still has a decreasing trend in both cases. This result

suggests that the CoD in each cluster is much more serious than that for the entire data set, which potentially makes the clusters easier to be separated in high-dimensional feature space.

Scenario 2: Suppose that we have a data set X with N = 2000 observations and n features. Observations in the first feature of X are composed of two equal-sized clusters with the Gaussian distribution, with the cluster centers as $\text{center}1 = 0$ and $\text{center}2 = 8$, the variance of each cluster equal to 1. For the remaining n-1 features, observations satisfy the Gaussian distribution with $\text{center} = [0, 0, ..., 0]_{1\times n\text{-}1}$ and covariance matrix as the n-1 dimensional identity matrix times the constant *variance* = 1. In this case, only the first feature makes contribution to clusters in the data, other features serve as the noisy information. We set n as 1, 100, and 1000, respectively, and show the histogram of the distance between points and origin in Fig. 5. This scenario suggests, as the feature dimensionality rises, clusters tend to merge together. Distributions of the intra cluster and inter cluster distances are given in Fig. 6, where we see distribution of the inter cluster distances is approaching that of the intra cluster distances.

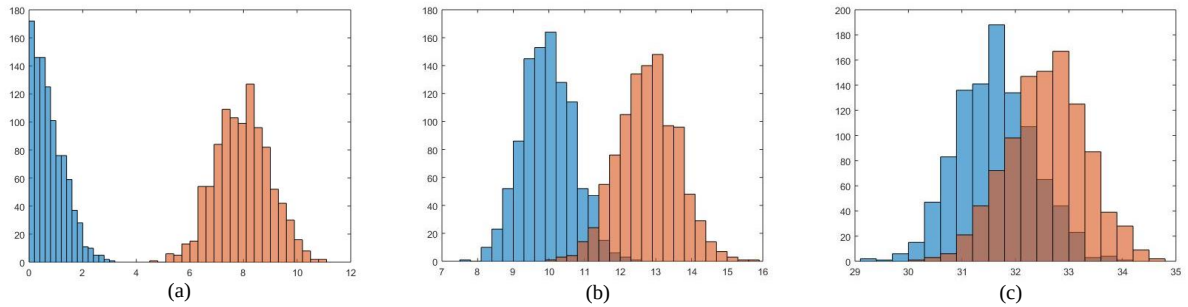


Fig. 5. Histogram of the distances to the origin when (a) n = 1; (b) n = 100; and (c) n = 1000.

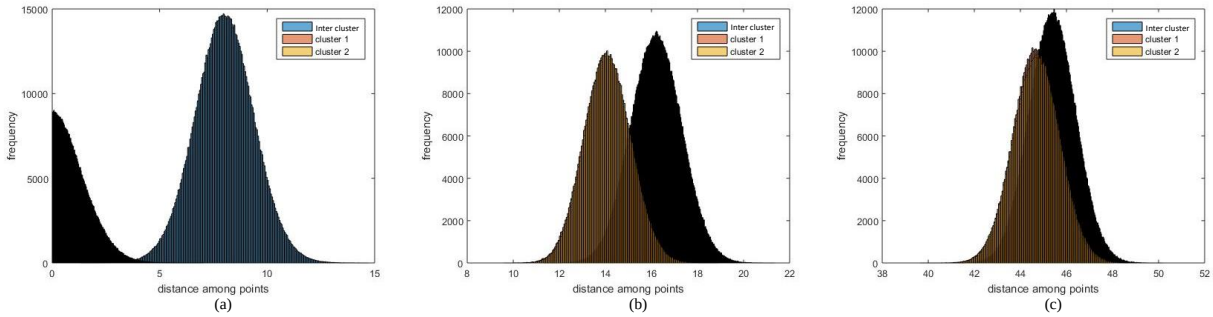


Fig. 6. Histogram of the inter cluster and intra cluster distances when (a) n = 1; (b) n = 100; and (c) n = 1000.

The indexes of the CoD of the whole data set and each cluster are also given in Fig. 7. In this scenario, both the cluster and the whole data set get a high concentration degree because the values of their relative variances are quite close to 0 with the increasing feature dimensionality.

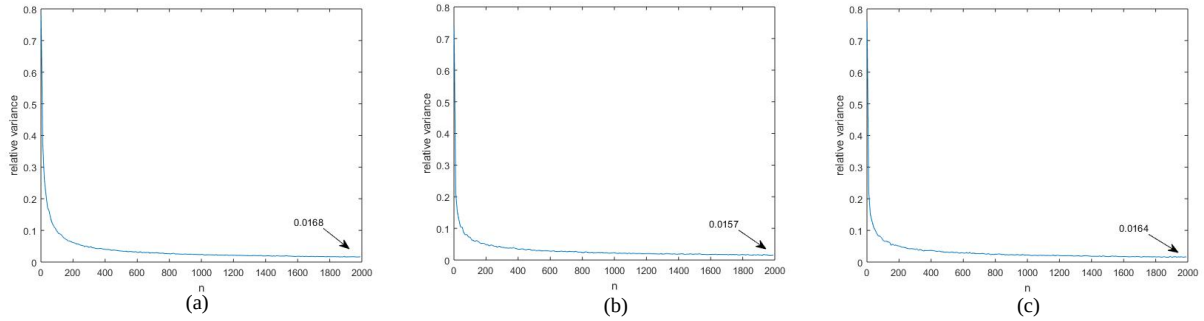


Fig. 7. Concentration degree for the whole data set and two clusters.

Scenario 3: Now let us see a data set with highly correlated features. The construction of the data set is similar to that in Scenario 1, X is composed of two equal-sized clusters satisfying the multivariable Gaussian distribution, with the centers as $\text{center1} = [0, 0, ..., 0]_{1\times n}$ and $\text{center2} = [8, 8, ..., 8]_{1\times n}$, the covariance matrix is the same as the one in Scenario 1, except that all non-diagonal entries has a value of 0.9. This setting indicates that features of the data are highly linear correlated. Similarly, we show the 3 groups of results in Figs. 8, 9, and 10.

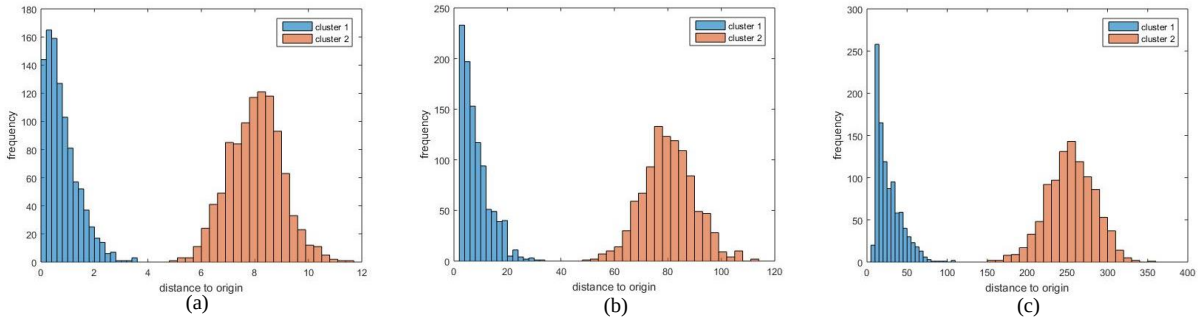


Fig. 8. Histogram of the distances to the origin when (a) n = 1; (b) n = 100; and (c) n = 1000.

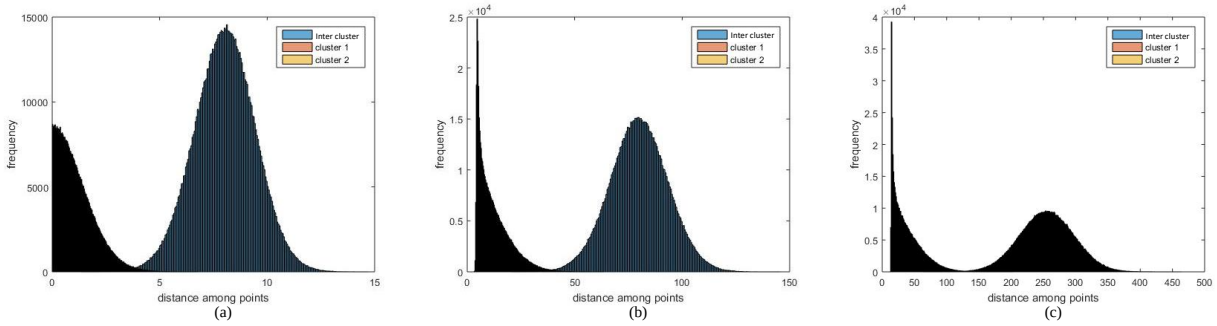


Fig. 9. Histogram of the inter cluster and intra cluster distances when (a) n = 1; (b) n = 100; and (c) n = 1000.

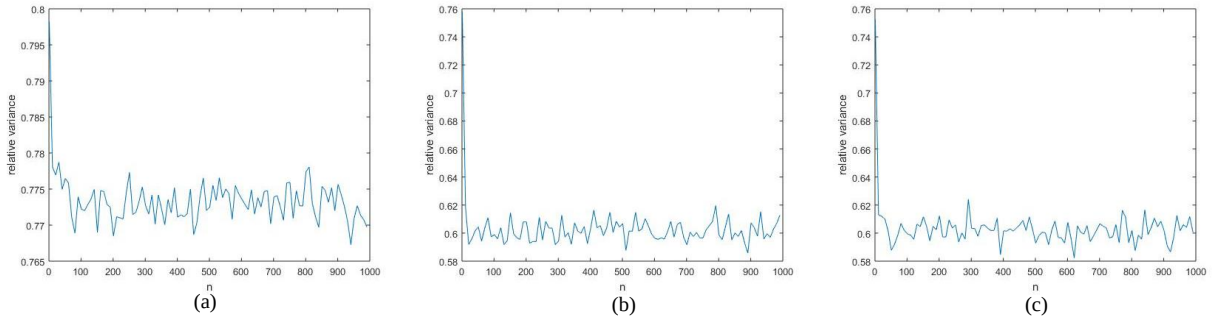


Fig. 10. Concentration degree for the whole data set and two clusters.

In this scenario, we see that with the increasing dimensionality, the clusters are always separable. In fact, from Fig. 10 since the relative variance is very large (> 0.5) and does not have a trend of decreasing to 0, we can say that the CoD phenomenon does not happen in this case.

From the current experimental results of the 3 scenarios, it is fair to conclude that high intrinsic dimension is beneficial for distinguishing clusters because the CoD occurred in this case is helpful. However, keep in mind that high intrinsic dimension barely exists for real-world data sets. Features in real-world data sets tend to be noisy and correlated with each other. This makes the clusters contained in the intrinsic dimension merged together with the increasing feature dimensionality (results in Scenario 2), or repeatedly represented in the increasing feature space (results in Scenario 3), which greatly increases the computing burden for clustering but is unnecessary at all. These findings motivate us to choose a low number of (yet informative) features when performing a clustering task in real world. Note that it is not the occurrence of the CoD that prompts to reduce the feature dimensionality, but the existence of noisy and correlated features (which makes the clustering non-effective and inefficient).

3. Clustering based on Autoencoder

An autoencoder [25] is a type of feedforward neural network which is trained to replicate its input at its output. Its structure is illustrated in Fig. 11, which is composed of the encoder part and the decoder part. In the encoder part, the values of the inputs are first linearly combined and sent to a hidden neuron in the hidden layer, then a nonlinear function is used to further encode the combination. In the decoder part, the output of these hidden neurons are linearly combined and sent to the output neurons. The weights of the connections between the neurons are adjusted such that the errors between the inputs and outputs are minimized. The crux of the autoencoder is that the inputs are finally represented by a small number of hidden neurons, which contains the most important and relevant information of the data set. Since the inception of the autoencoder, many different versions of the network have been proposed to make it more robust. The one we use in this study is enhanced by the L_2 and sparse regularization, whose cost function of the network is represented as

$$E = \frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{N}(x_{kj} - \hat{x}_{kj})^2 + \lambda\Omega_{\text{weights}} + \beta\Omega_{\text{sparsity}} \tag{5}$$

where λ is the coefficient for the L_2 regularization term and β is the coefficient for the sparsity regularization term. One may find the detailed formulas used for two regularization items in [26]. Given the dimensionality reduction an autoencoder enables to serve, it is therefore utilized in this paper to reduce the high-dimensional feature space to mitigate the issue associated with FCM as discussed in the preceding section.

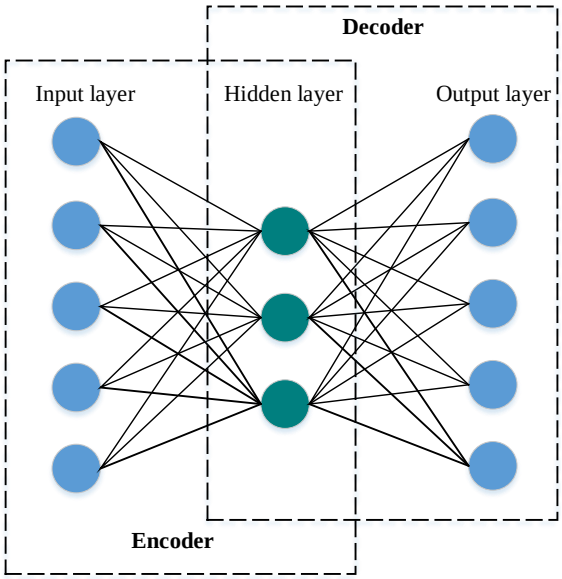
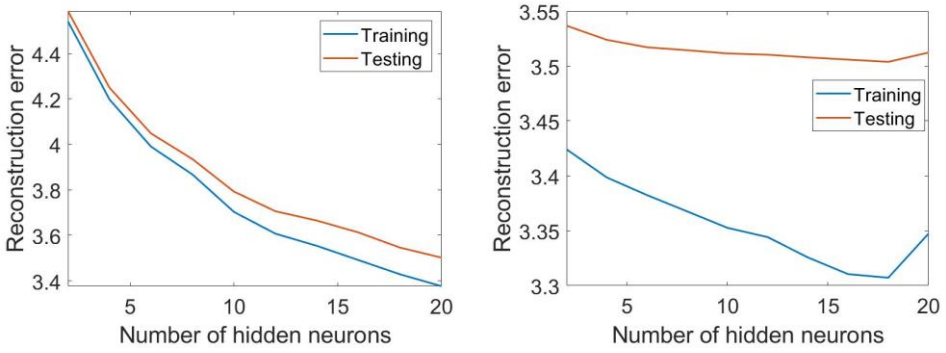


Fig. 11 Structure of the autoencoder.

4. Experimental Studies

In this section, we apply the proposed strategy to the two UCI data sets mentioned in the introduction. As for the experimental settings, 60% of the original data points are used as the training data, while the remaining as the testing data. We use the *trainAutoencoder* function provided in MATLAB to train this network, and the parameters of this function are remained as the default setting, except that the number of hidden neurons is ranged from 2 to 20 with a stepsize of 2. We record the changing trend of the reconstruction error E with respect to this number. Then we focus on a selected number of hidden neurons, which serve as the inputs to FCM for further analysis.

First, we show the performance of the autoencoder with respect to the different number of hidden neurons in Fig. 12. Generally, with a larger number of hidden neurons, small reconstruction error is observed. However, the decreasing trend tends to be slow down with the increasing number of hidden neurons, and we can even see that the reconstruction error tends to be rebound back when this number is large for data set Hand. Hence, for both data sets let us focus on the scenario where only 6 neurons are considered, that is only 6 new features of each data set are used. We only show the clustering results related to the prototypes in Table 2. Clearly, now for both data sets the obtained prototypes are distinguishable with each other.



(a) (b)

Fig. 12. Trends of the reconstruction error of data sets (a) Isolet and (b) Hand.

Table 2 Results of the new features of the obtained prototypes.

Data Sets	Prototypes	Features					
		f_1	f_2	f_3	f_4	f_5	f_6
Isolet	$\boldsymbol{v}_1$	0.17	0.24	0.29	0.16	0.19	0.13
	$\boldsymbol{v}_2$	0.36	0.16	0.14	0.24	0.51	0.28
Hand	$\boldsymbol{v}_1$	0.22	0.24	0.23	0.28	0.20	0.25
	$\boldsymbol{v}_2$	0.05	0.04	0.07	0.05	0.03	0.12

5. Conclusion

In this paper, we first pointed out the problem faced by the FCM when this clustering algorithm is performed in high-dimensional feature space, which potentially results in indistinguishable prototypes. A detailed analysis of the reason for this failure is carefully discussed under several different scenarios. We highlighted that the well-known concentration of distance (CoD) may not necessarily lead to a bad performance of the clustering algorithm; rather it is the noisy and redundant (correlated) features that could lead to the poor performance. Hence, we applied the autoencoder, a powerful dimensionality reduction technique, to seek for the most relevant features (newly formed features) contributing to the structure of the data. The experimental results demonstrate the effectiveness of the autoencoder in supporting FCM generating well separated prototypes.

Note that many state-of-the-art methods [27] have been proposed to cluster the high-dimensional data with FCM, say, those based on sparse regularity [28] and unsupervised feature selection [29]. As a future study, it is interesting to compare the proposed method in this study with those in the current literature. Besides, since FCM is a popular building block for some other system modeling techniques, say, the fuzzy rule-based model [30]–[32], in the future study we intend to research how the autoencoder could help to improve the performance (e.g., accuracy) of the prediction model.

References

[1] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[2] J. C. Dunn, "Well-separated clusters and optimal fuzzy partitions," *J. Cybern.*, vol. 4, no. 1, pp. 95–104, 1974.

[3] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, no. 2–3, pp. 191–203, 1984.

[4] N. Päivinen, "Clustering with a minimum spanning tree of scale-free-like structure," *Pattern Recognit. Lett.*, vol. 26, no. 7, pp. 921–930, 2005.

[5] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 11, pp. 1101–1113, 1993.

[6] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," *Comput. J.*, vol. 26, no. 4, pp. 354–359, 1983.

[7] G. Karypis, E.-H. S. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," *Computer (Long. Beach. Calif).*, no. 8, pp. 68–75, 1999.

[8] H. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 1, no. 3, pp. 231–240, 2011.

[9] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, 1996, vol. 96, no. 34, pp. 226–231.

[10] Y. Shen and W. Pedrycz, "Collaborative fuzzy clustering algorithm: Some refinements," *Int. J. Approx. Reason.*, vol. 86, pp. 41–61, 2017.

[11] Y. Shen, W. Pedrycz, and X. Wang, "Clustering homogeneous granular data: Formation and evaluation," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1391–1402, 2019.

[12] Y. Shen, W. Pedrycz, Y. Chen, X. Wang, and A. Gacek, "Hyperplane Division in Fuzzy C-Means: Clustering Big Data," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 11, pp. 3032-3046, 2020.

[13] L. A. Zadeh, "Fuzzy Sets-Information and Control-1965," *Inf. Control*, 1965.

[14] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.

[15] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is 'nearest neighbor' meaningful?," in *International conference on database theory*, 1999, pp. 217–235.

[16] D. François, V. Wertz, and M. Verleysen, "The concentration of fractional distances," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 7, pp. 873–886, 2007.

[17] S. Kumari and B. Jayaram, "Measuring concentration of distances—an effective and efficient empirical index," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 2, pp. 373–386, 2016.

[18] C.-M. Hsu and M.-S. Chen, "On the design and applicability of distance functions in high-dimensional data space," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 4, pp. 523–536, 2008.

[19] V. Pestov, "Is the k-NN classifier in high dimensions affected by the curse of dimensionality?," *Comput. Math. with Appl.*, vol. 65, no. 10, pp. 1427–1437, 2013.

[20] A. K. Pal, P. K. Mondal, and A. K. Ghosh, "High dimensional nearest neighbor classification based on mean absolute differences of inter-point distances," *Pattern Recognit. Lett.*, vol. 74, pp. 1–8, 2016.

[21] F. Klawonn, F. Höppner, and B. Jayaram, "What are clusters in high dimensions and are they difficult to find?," in *International workshop on clustering high-dimensional data*, 2012, pp. 14–33.

[22] E. Levina and P. J. Bickel, “Maximum likelihood estimation of intrinsic dimension,” in *Advances in neural information processing systems*, 2005, pp. 777–784.
[23] M. Radovanovic, A. Nanopoulos, and M. Ivanovic, “Hubs in space: Popular nearest neighbors in high-dimensional data,” *J. Mach. Learn. Res.*, vol. 11, no. sept, pp. 2487–2531, 2010.
[24] R. J. Durrant and A. Kabán, “When is ‘nearest neighbour’meaningful: A converse theorem and implications,” *J. Complex.*, vol. 25, no. 4, pp. 385–397, 2009.
[25] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science (80-.).*, vol. 313, no. 5786, pp. 504–507, 2006.
[26] B. A. Olshausen and D. J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by V1?,” *Vision Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.
[27] Z. Deng, K.-S. Choi, Y. Jiang, J. Wang, and S. Wang, “A survey on soft subspace clustering,” *Inf. Sci. (Ny).*, vol. 348, pp. 84–106, 2016.
[28] X. Chang, Q. Wang, Y. Liu, and Y. Wang, “Sparse Regularization in Fuzzy c-Means for High-Dimensional Data Clustering,” *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2616–2627, 2016.
[29] P. Mitra, C. A. Murthy, and S. K. Pal, “Unsupervised feature selection using feature similarity,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 301–312, 2002.
[30] Y. Shen, W. Pedrycz, X. Jing, A. Gacek, X. Wang, and B. Liu, “Identification of Fuzzy Rule-Based Models with Output Space Knowledge Guidance,” *IEEE Trans. Fuzzy Syst.*, 2020.
[31] X. Hu, Y. Shen, W. Pedrycz, Y. Li, and G. Wu, “Granular Fuzzy Rule-Based Modeling With Incomplete Data Representation,” *IEEE Trans. Cybern.*, 2021.
[32] T. Chen, C. Shang, J. Yang, F. Li, and Q. Shen, “A New Approach for Transformation-based Fuzzy Rule Interpolation,” *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 12, pp. 3330-3344, 2019.