Fuzzy Separation And Shrinkage Clustering

Ning Yu, Yanxiang Zong, Haoran Liang, Heying Zhu, Kunhong Liu, Qingqiang Wu*
School of Informatics, Xiamen University, Xiamen 361005, China
wuqq@xmu.edu.cn

Abstract. Clustering has many applications in data mining and machine learning. Fuzzy clustering methods have been widely used in clustering. However, fuzzy clustering methods still have a fatal problem: the cluster radius sensitivity problem. The cluster radius sensitivity problem means that clusters with smaller radius will predominate in clustering and obtain more data points. Aiming at this problem, we propose a fuzzy separation and shrinkage clustering algorithm (FSC). FSC uses cluster membership degrees and cluster sizes to construct a new membership distribution, and then moves the data points according to this new membership distribution. The accuracies of our algorithm on wine, iris, balance scale and seeds are as follows: 98.82%, 97.27%, 63.07% and 91.34%. Our contributions are: (1) We propose a fuzzy separation and shrinkage clustering algorithm, which can solve the cluster radius sensitivity problem. (2) The performance of our algorithm on the UCI datasets goes beyond the benchmark algorithms.

1. Introduction
Clustering is an important method in data mining, which has been widely used in biology, economy, agriculture and so on [1][2][3]. Clustering divides data into different clusters according to similarity without supervision. In order to solve the category uncertainty problem of edge data points, fuzzy clustering is proposed. Zadeh proposed the concept of fuzzy sets in 1965 [4]. In 1969, Ruspini proposed the fuzzy partition theory, changing the membership range of hard partitioning from \{0, 1\} to \[0,1\], and replacing the clear set with the fuzzy set [5]. Dunn proposed a fuzzy c-means clustering model (FCM) in 1974 [6]. In 1981, Bezdek extended the fuzzy clustering model by adding a weighted index of fuzzy membership degrees [7]. In order to solve the problem of noise data and outliers, PCM [8] is proposed, which removes the restriction that each individual belongs to each class in a probability sum of 1. In 2003, Fan proposed a suppression fuzzy c-means clustering algorithm (S-FCM), which links hard c-means (HCM) and FCM by highlighting the main factors and suppressing secondary factors [9]. FCM is the most widely known algorithm in fuzzy clustering. It changes the category assignment of hard clusters. Each sample can belong to a category with 0-1. Compared with HCM, FCM improves the clustering performance and more rationally reveals the data distribution. FCM has been researched and applied in many aspects, and many variants have been proposed [10][11][12]. However, fuzzy clustering methods still have a fatal problem: the cluster radius sensitivity problem. The cluster radius sensitivity problem means that clusters with smaller radius will predominate in clustering and obtain more data points. As shown in Fig. 1, because the radius of cluster A is smaller than cluster B, fuzzy clustering algorithms will divide the points that originally belong to B to A. The reason for this problem is that fuzzy clustering algorithms tend to generate spherical clusters. Aiming at this problem, this paper proposes a fuzzy separation clustering algorithm. In order to punish clusters with small radius, this paper uses cluster membership degrees and cluster sizes to construct a new
membership distribution, and then moves the data points according to this new membership distribution. The effect this paper wants to achieve is shown in Fig. 2. Through cluster separation and shrinkage, the radii of A and B is the same, so that the fuzzy clustering method can be applied. Our contributions are: (1) We propose a fuzzy separation and shrinkage clustering algorithm, which can solve the cluster radius sensitivity problem. (2) The performance of our algorithm on the UCI datasets goes beyond the benchmark algorithms.

2. Method

In this paper we propose a fuzzy separation clustering algorithm based on cluster attraction. Separation means that edge data points are attracted to different clusters and moved toward the center of each cluster. As shown in Fig. 2, this is a process of shrinking within a cluster and separating between clusters. FSC can be divided into two phases: initial clustering and separation and contraction. The process of FSC is shown in Algorithm 1. There are two issues that need to consider here: how to make points move and how to ensure that the move meets the requirements?

1) How to make points move: considering the concept of movement, we need an initial state and a target state. We take the membership probability distribution \{μij|1 ≤ i ≤ c, 1 ≤ j ≤ N\} as the initial state. Then we assume that there is a better target state \{qij|1 ≤ i ≤ c, 1 ≤ j ≤ N\} than the initial state. KL divergence is a measure of the difference between two distributions. For two probability distributions P and Q, \(KL(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}\). We use \(KL(\mu||q) = \sum_i \sum_j \mu_{ij} \log \frac{\mu_{ij}}{q_{ij}}\) as the distance between the initial state and the target state. It should be noted that \(KL(\mu||q)\) is a function of \{xj|1 ≤ j ≤ N\} as an independent variable. Since the separation of edge data points is sensitive, we choose to buffer from the autoencoder, just like a spring, to avoid the separation process is too sharp. An autoencoder [13] is a neural network whose output is equal to the input: \(AE(X) = X\). We then minimize \(KL(\mu||q)\), which means that \{xj|1 ≤ j ≤ N\} will move to positions that match the target state.
Algorithm 1 FSC

Train an autoencoder AE, the parameter of AE is $\theta$.

Perform initial clustering to get: cluster center $\mu_i \mid 1 \leq i \leq c$, cluster set $\{C_i \mid 1 \leq i \leq c\}$

$$h_{ij}^{ini} = \frac{(1+d_{ij})^{-1}}{\sum_{k=1}^{N}(1+d_{kj})^{-1}}$$

$$g_i = \sum_{j=1}^{N} h_{ij}^{ini}$$

$$\text{reg}_{i} = \frac{g_i}{\sum_{k=1}^{c} g_k} \left| C_{ini}^k \right|$$

for $s=1; s<\text{steps}; s++$ do

$$d_{ij} = ||AE(v_i^{ini}) - AE(x_j)||$$

$$h_{ij}^{ini} = \frac{(1+d_{ij})^{-1}}{\sum_{k=1}^{N}(1+d_{kj})^{-1}}$$

$$f_i = \sum_{j=1}^{N} h_{ij}$$

$$\text{reg}_{i} = \frac{g_i}{\sum_{k=1}^{c} g_k} \left| C_{ini}^k \right|$$

Gradient update: $\nabla \theta_{,\mu} KL(\mu, q)$

end for

2) How to ensure that the move meets the requirements: after the initial clustering, we will get the divided clusters $\{C_i \mid 1 \leq i \leq c\}$ and cluster centers $\{\mu_i \mid 1 \leq i \leq c\}$. We define:

$$\text{reg}_{i} = \frac{g_i}{\sum_{k=1}^{c} g_k} \left| C_{ini}^k \right|$$ (1)

where $g_i = \sum_{j=1}^{N} h_{ij}^{ini}, h_{ij}^{ini} = \frac{(1+d_{ij})^{-1}}{\sum_{k=1}^{N}(1+d_{kj})^{-1}}, d_{ij} = ||v_i^{ini} - x_j||$ and $\left| C_{ini}^k \right|$ is the number of objects in cluster $C_{ini}^k$. Then we define $q_{ij}$ as follows:

$$q_{ij} = \frac{\mu_{ij}/f_i}{\sum_{k=1}^{c} g_k} \text{reg}_{i}$$ (2)

where $f_i = \sum_{j=1}^{N} h_{ij}, h_{ij} = \frac{(1+d_{ij})^{-1}}{\sum_{k=1}^{N}(1+d_{kj})^{-1}}, d_{ij} = ||v_i - x_j||$ and $t$ is a weighting parameter that controls the amount of separation and shrink.

Now we will discuss how $q_{ij}$ meets the requirements. As shown in Fig. 1, we define $r(A)$ and $r(B)$ as the radii of A and B, $|A|$ and $|B|$ as the number of objects in cluster A and cluster B and $|C_{ini}^A|$ and $|C_{ini}^B|$ as the number of objects in $C_{ini}^A$ and $C_{ini}^B$ after the initial clustering. Then, we discuss the effectiveness of the $q_{ij}$ in two ways:

(1)$|A| \geq |B|$

When $|A| \geq |B|$, we have:

$$|C_{ini}^A | > |C_{ini}^B |$$

$$g_A > g_B$$

$$\frac{1}{f_A} < \frac{1}{f_B}$$

So we can punish A with $\frac{1}{f_A}$. In order to avoid excessive punishment (considering that in the separation process, $\frac{1}{f_A}$ will become smaller and smaller), we use $\text{reg}_{A}$ as the regular term. As shown in
Fig 2, under the interaction of $\frac{1}{f_A}$ and $reg_A$, the algorithm performs the process of shrinking within a cluster and separating between clusters, and finally clusters the incorrectly divided data correctly.

(2) $|A| < |B|$

When $|A| < |B|$, we have:

$$|C_A^{ini}| < |C_B^{ini}|$$

(6)

$$g_A < g_B$$

(7)

$$\frac{1}{f_A} > \frac{1}{f_B}$$

(8)

So we can reward B with $reg_B$. In order to avoid over-reward (considering that the $\frac{1}{f_B}$ of the cluster B will become larger and larger during the separation process), we can use $\frac{1}{f_B}$ as a regular term. As shown in Fig 2, under the interaction of $\frac{1}{f_B}$ and $reg_B$, the algorithm performs the process of shrinking within a cluster and separating between clusters, and finally clusters the incorrectly divided data correctly.

3. Experiment

3.1. Datasets

In order to verify the effect of the algorithm, we selected four datasets with higher usage rates on the UCI dataset for experiments. Table 1 shows the properties of these four datasets.

| Dataset   | Number of objects | Number of cluster | Dimension |
|-----------|-------------------|-------------------|-----------|
| Iris      | 150               | 3                 | 4         |
| Wine      | 178               | 3                 | 13        |
| Balance scale | 625           | 3                 | 5         |
| Seeds     | 210               | 3                 | 7         |

3.2. Evaluation Criteria

We use the accuracy (ACC) as the evaluation standard. The definitions of NMI and ARI are as follows:

$$ACC = \max_m \frac{\sum_{j=1}^n t_j=m(c_j)}{n}$$

(9)

where $n$ is the number of all samples, $t_j$ is the ground-truth label, $c_j$ is the clustering result, and $m$ is all possible one-to-one mappings between labels and clusters.

3.3. Result

Table 2 shows the performance of the algorithm on ACC. We run each algorithm 10 times and then take the evaluation standard average. For different $t$, the best performance is when $t=2$, which is FSC in the Table 2. The algorithm performs better when using an autoencoder, which shows the spring effect of autoencoder.

|          | Wine  | Iris  | Balance scale | Seeds  |
|----------|-------|-------|---------------|--------|
| FCM      | 0.9494| 0.8933| 0.5209        | 0.9000 |
| FSC without AE | 0.9663| 0.9200| 0.5539        | 0.8048 |
| FSC$_{t=1}$ | 0.9567| 0.9333| 0.5219        | 0.9017 |
| FSC$_{t=3}$ | 0.9556| 0.9333| 0.5214        | 0.9086 |
| FSC      | 0.9882| 0.9727| 0.6307        | 0.9134 |
4. Conclusion
Fuzzy clustering methods have a fatal problem: the cluster radius sensitivity problem. The cluster radius sensitivity problem means that clusters with smaller radius will predominate in clustering and obtain more data points. Aiming at this problem, this paper proposes a fuzzy separation clustering algorithm. In order to punish clusters with small radius, this paper uses cluster membership degrees and cluster sizes to construct a new membership distribution, and then moves the data points according to this new membership distribution. The performance of our algorithm on the UCI dataset goes beyond the benchmark algorithm. At the same time, our algorithm is still sensitive to the initial clustering centers and the separation process. In future research, we will continue to improve the robustness of the algorithm.

Acknowledgments
This work was supported in part by the National Natural Science Foundation of China under Grant No. 61502402, and the Fundamental Research Funds for the Central Universities under Grant No. 20720180073.

References
[1] Ahmad A, Khan S S. Survey of state-of-the-art mixed data clustering algorithms[J]. IEEE Access, 2019, 7: 31883-31902.
[2] Xu R, Wunsch D. Clustering[M]. John Wiley & Sons, 2008.
[3] Jain A K. Data clustering: 50 years beyond K-means[J]. Pattern recognition letters, 2010, 31(8): 651-666.
[4] Zadeh L A. Fuzzy sets[J]. Information and control, 1965, 8(3): 338-353.
[5] Ruspini E H. A new approach to clustering[J]. Information and control, 1969, 15(1): 22-32.
[6] Dunn J C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters[J]. 1973
[7] Bezdek J C. Pattern recognition with fuzzy objective function algorithms[M]. Springer Science & Business Media, 2013.
[8] Krishnapuram R, Keller J M. A possibilistic approach to clustering[J]. IEEE transactions on fuzzy systems, 1993, 1(2): 98-110.
[9] Fan J L, Zhen W Z, Xie W X. Suppressed fuzzy c-means clustering algorithm[J]. Pattern Recognition Letters, 2003, 24(9-10): 1607-1612.
[10] Cannon R L, Dave J V, Bezdek J C. Efficient implementation of the fuzzy c-means clustering algorithms[J]. IEEE transactions on pattern analysis and machine intelligence, 1986 (2): 248-255.
[11] Kamel M S, Selim S Z. New algorithms for solving the fuzzy clustering problem[J]. Pattern recognition, 1994, 27(3): 421-428.
[12] Park D C, Dagher I. Gradient based fuzzy c-means (GBFCM) algorithm[C]//Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN’94). IEEE, 1994, 3: 1626-1631.
[13] Ng A. Sparse autoencoder[J]. CS294A Lecture notes, 2011, 72(2011): 1-19.