



Applying K-Means Clustering and Fuzzy C-Means Clustering in Vehicle Crashes

Azad Abdulhafedh

University of Missouri, Columbia, Missouri, USA

Email: dr.azad.s.a@gmail.com

How to cite this paper: Abdulhafedh, A. (2025) Applying K-Means Clustering and Fuzzy C-Means Clustering in Vehicle Crashes. *Open Access Library Journal*, 12: e12856. <https://doi.org/10.4236/oalib.1112856>

Received: December 23, 2024

Accepted: April 5, 2025

Published: April 8, 2025

Copyright © 2025 by author(s) and Open Access Library Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Clustering is an unsupervised machine learning technique used to organize unlabeled data into groups based on similarity. This paper applies the K-means and Fuzzy C-means clustering algorithms to a vehicle crash dataset in order to explore various patterns in the data. K-means assigns data points to clusters based on the similarity between the data point and the cluster centroids, which results in partitioning the data into distinct clusters. On the other hand, fuzzy C-means clustering allows data points to belong to multiple clusters simultaneously with varying degrees of membership, providing a more diverse representation of the data. Results show that while K-means clustering is simpler and easier to interpret, fuzzy C-means clustering offers more flexibility and can manage situations where data points may have more cluster assignments.

Subject Areas

Applications of Communication Systems

Keywords

Clustering, K-Means, Fuzzy C-Means, Vehicle Crashes, Dunn's Partition Coefficient, Machine Learning

1. Introduction

In machine learning, the process of dividing datasets into groups consisting of similar data points is called clustering, which is an unsupervised learning technique [1]-[3]. The data should be unlabeled, meaning we do not have a dependent variable (or response variable) for the algorithm to compare as a basic truth. Instead, we try to split the dataset into diverse groups so that the data points in the same group have characteristics similar to those of the data points in separate groups. Clustering algorithms do not depend on common assumptions that the

conventional statistical methods use, such as the underlying distribution of data, and therefore they are useful in situations where little prior knowledge exists [4]-[6]. Clustering can reveal underlying patterns in data in many applications, including classification, image processing, stock market analysis, pattern recognition, customer segmentation, and identification. Types of clustering include exclusive clustering, in which data points exclusively belong to one cluster, such as K-means clustering. Another type is the overlapping clustering, in which data points belong to multiple clusters, such the Fuzzy C-means clustering. Another type is Hierarchical clustering that includes grouping similar objects into clusters, in which each cluster is distinct from another cluster and the object within that cluster are similar to each other [7] [8]. This paper incorporates the K-means and Fuzzy C-means clustering in vehicle crashes.

2. K-Means

K-means is a widely used clustering algorithm due to its simplicity and clarity. K-Means was originally discovered by Polish Mathematician Hugo Steinhaus in 1956. However, the term “K-means” was first used by James MacQueen in 1967. Later, David Arthur and Sergei Vassilvitskii produced k-means augmentation in 2007 [9]-[13]. The K value represents the number of clusters, and the algorithm works by randomly assigning centroids to the dataset. Then, clusters can be formed by assigning data points to the nearest cluster. From those clusters, new centroids are formed with the mean data points. This iterative process continues until convergence is achieved by finding the optimal clustering results. The algorithm’s efficiency relies on defining appropriate distance measures, such as Euclidean distance, to determine similarity between data points. K-means uses a centroid-based method, iteratively updating cluster centers to minimize intra-cluster distances, which minimizes the Within-Cluster Sum of Squares, enhancing the accuracy of clustering results. Some limitations of K-means include its sensitivity to outliers, affecting the centroids’ positioning, which can produce suboptimal clustering results. It also struggles with non-linear relationships between data points, affecting the accuracy of cluster outputs. In addition, the K-means algorithm does not work with categorical data [14]-[17].

In order to understand the math behind the K-means, we need to know that each cluster is represented by its center (*i.e.*, centroid) which corresponds to the mean of the observation values assigned to the cluster. In addition, K-means clustering, uses the sum of Euclidean distances from points to their nearby cluster centroids. The formula for Euclidean distance is given by [18]-[20]:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

The objective function for K-means is given by:

$$J = \sum_{i=1}^m \sum_{k=1}^m w_{ik} \|x_i - c_k\|^2$$

where:

$w_{ik} = 0$ if the data point does not belong to the cluster, and

$w_{ik} = 1$ if the data point belongs to the cluster

J : objective function

k : number of clusters

m : number of cases

X_i : case i

c : centroid of cluster k

Choosing the optimal number of clusters is a critical step in K-means, and since we don't have predefined cluster counts in unsupervised learning, we need a systematic approach to determine the best k value. The Elbow Method is widely used for this purpose in K-Means clustering. We iterate over a range of k values, typically from 1 to n (where n is a user specified hyper-parameter that you choose). Then, for each k , we calculate the Within-Cluster Sum of Squares, which is the sum of squared distances between each member of the cluster and its centroid. Then, the K value is chosen after the decrease of these distances is almost constant [21]-[23].

3. Fuzzy C-Means

Fuzzy C-means is a clustering algorithm that can manage overlapping clusters. It is considered a soft clustering technique in which every data point is assigned to a cluster along with the probability of it being in the cluster. Therefore, it assigns each data point a degree of membership or probability that indicates the likelihood of a data point belonging to each cluster. Thus, it allows data points to belong to multiple clusters [24]-[29]. It works by randomly initializing cluster centroids from the data set and specifying a fuzziness parameter (m) to control the degree of fuzziness in the clustering. Then, it calculates the probability of each data point to each cluster based on its distance to the cluster centroids using a distance metric, such as Euclidean distance. The process is iterative until a specified number of iterations is reached, or the membership values and centroids converge to stable values. This algorithm is less sensitive to outliers as it allows for soft, probabilistic cluster assignments. The math behind Fuzzy C-means includes [27]-[30]:

Our goal is to minimize the objective function which is as follows:

$$J_m = \sum_{i=1}^n \sum_{j=1}^c w_{ij}^m \|\mathbf{x}_i - \mathbf{v}_j\|^2$$

where:

n = number of data point

c = number of clusters

x = " i " data point

v = centroid of " j " cluster

w = membership value of data point of i to cluster j

m = fuzziness parameter ($m > 1$)

Then, we need to update the membership values using the formula:

$$w_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{\|\mathbf{x}_i - \mathbf{v}_j\|}{\|\mathbf{x}_i - \mathbf{v}_k\|} \right)^{\frac{2}{m-1}}}$$

Then, we need to update cluster centroids using a weighted average of the data points:

$$\mathbf{v}_j = \frac{\sum_{i=1}^n w_{ij}^m \cdot \mathbf{x}_i}{\sum_{i=1}^n w_{ij}^m}$$

Then, we keep updating the membership values and the cluster centers until the membership values and cluster centers stop changing significantly or when a pre-defined number of iterations is reached. This is followed by assigning each data point to the cluster or multiple clusters for which it has the highest membership value.

In order to choose the appropriate number of clusters in Fuzzy C-means, Dunn's partition coefficient can be used, which measures how close the fuzzy solution is to the corresponding K-means solution. Dunn's partition coefficient is a clustering measure that quantifies the degree of separation between different clusters. The formula for Dunn's partition coefficient is [28] [29]:

$$F(U) = \frac{1}{N} \sum_{k=1}^K \sum_{i=1}^N m_{ik}^2$$

where m_i represents the membership value of each object to each cluster, and n is the total number of observations. This coefficient ranges from $1/K$ to 1. Its value is $1/K$ when all memberships are equal to $1/K$. Dunn's partition coefficient may be normalized so that it varies from 0 (completely fuzzy) to 1 (hard cluster). The normalized version is:

$$F_c(U) = \frac{F(U) - (1/K)}{1 - (1/K)}$$

Another partition coefficient, given in Kaufman [30], is the $D(U)$. This coefficient ranges from 0 (hard clusters) to $1 - 1/K$ (completely fuzzy). The normalized version of this coefficient is [30]:

$$D_c(U) = \frac{D(U)}{1 - (1/K)}$$

$F_c(U)$ and $D_c(U)$ together can be used as a perfect indication of an optimum number of fuzzy C clusters. We should choose C so that $F_c(U)$ is large, and $D_c(U)$ is small.

4. Data

The data used in this paper presents the number of vehicle crashes at two local streets A and B in the city of Columbia, Missouri, USA from the years 2008 to 2019. The data includes three columns namely, "Year", "Number of crashes at street A", and "Number of crashes at street B" as shown in **Table 1**.

Table 1. Number of vehicle crashes at street A and B from 2008-2019.

| Year | Number of crashes at street A | Number of crashes at street B |
|------|-------------------------------|-------------------------------|
| 2008 | 61 | 52 |
| 2009 | 53 | 38 |
| 2010 | 74 | 42 |
| 2011 | 69 | 51 |
| 2012 | 81 | 44 |
| 2013 | 48 | 71 |
| 2014 | 59 | 52 |
| 2015 | 61 | 78 |
| 2016 | 47 | 56 |
| 2017 | 82 | 49 |
| 2018 | 74 | 68 |
| 2019 | 83 | 47 |

In order to determine the K value, the Elbow curve method was utilized using the NCSS software, and a $K = 5$ was chosen for both the K-means and C-means. **Figure 1** shows the number of vehicle crashes at street A and B from 2008 to 2019. We can see that the crashes at both streets for the years 2010, 2012, 2017, and 2019 are grouped as cluster 1. The crashes at both streets for the years 2013, and 2015 are grouped as cluster 2. The crashes at both streets for the years 2008, 2014, and 2016 are grouped as cluster 3. The crashes at both streets for the year 2009 are grouped as cluster 4. The crashes at both streets for the years 2011, and 2018 are grouped as cluster 5.

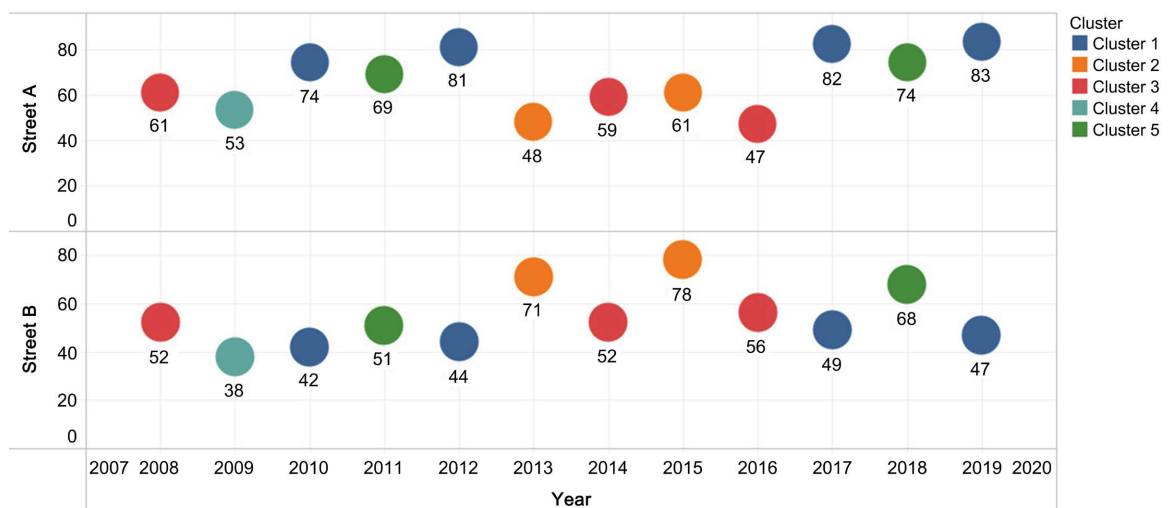
**Figure 1.** Number of vehicle crashes at street A and B from 2008-2019.

Figure 2 shows the trendlines (*i.e.*, increase, constant or decrease) of vehicle crashes at street A and B for different years. For example, the number of crashes

at street A in cluster 3 is decreasing from 61 in 2008 to 59 in 2014 to 47 in 2016, while it is constant at street B from 2008 to 2014, then it is increasing to 56 in 2016.

Figure 3 shows the clusters of crashes at street A vs street B.

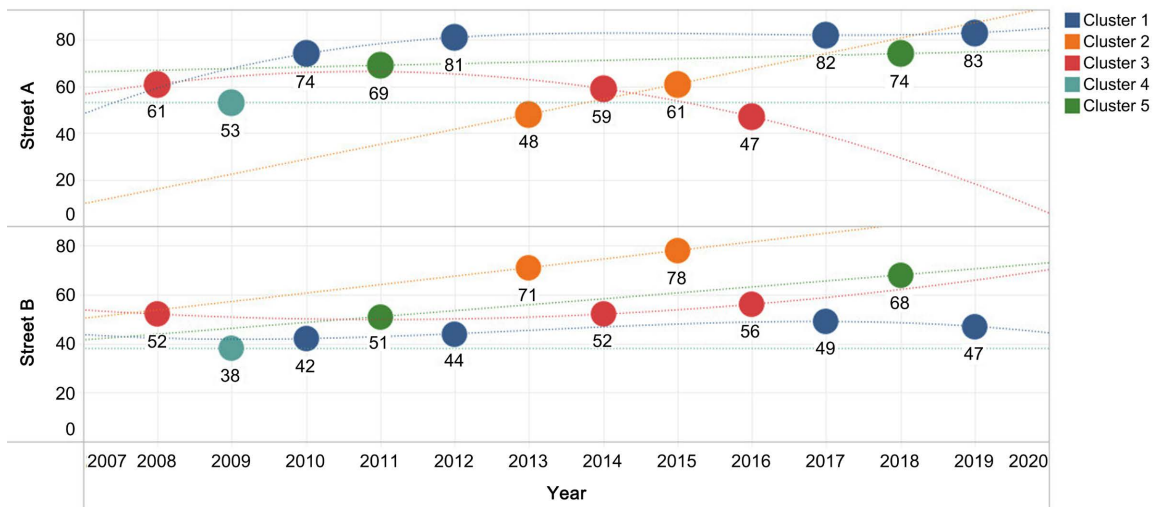


Figure 2. The trendlines of vehicle crashes at street A and B from 2008 to 2019.

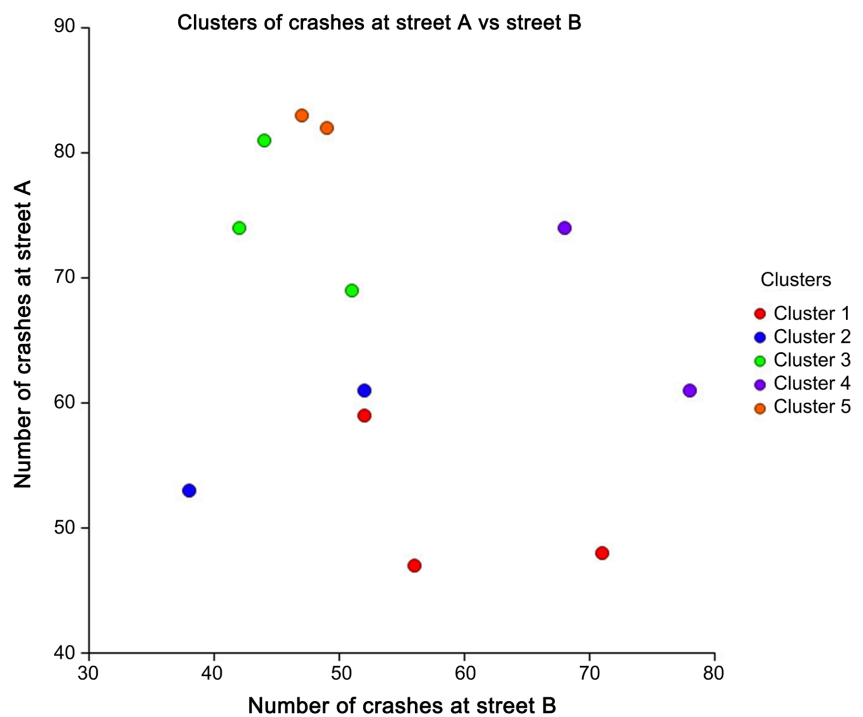


Figure 3. Clusters of vehicle crashes at street A vs street B.

The summary statistics for the dataset are shown in Table 2. We can see that the mean of crashes at street A is 66 and the standard deviation is 13.003. While the mean of crashes at street B is 54 and the standard deviation is 12.269. The minimum number of crashes at street A is 47 and the maximum number of crashes is 83. While the minimum number of crashes at street B is 38 and the

maximum number of crashes at street B is 78.

Table 2. Summary statistics of the dataset.

| Statistic | Crashes at Street A | Crashes at Street B |
|-----------------------------|---------------------|---------------------|
| Mean | 66 | 54 |
| Standard Deviation | 13.003 | 12.269 |
| Standard Error | 3.7537 | 3.5419 |
| Lower 95% CL Mean | 57.737 | 46.204 |
| Upper 95% CL Mean | 74.262 | 61.795 |
| Minimum | 47 | 38 |
| Maximum | 83 | 78 |
| Shapiro-Wilk Normality test | 0.9219 | 0.9693 |

The cluster's mean of each cluster at each street is shown in **Table 3**. For example, the mean of cluster 1 at street A is 51.33, while the mean of cluster 1 at street B is 59.66. The mean of cluster 3 at street A is 74.66, while the mean of cluster 3 at street B is 45.66.

Table 3. The cluster's mean at street A and B.

| Street | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 | Cluster 5 |
|--------|-----------|-----------|-----------|-----------|-----------|
| A | 51.33 | 57 | 74.66 | 67.5 | 82.5 |
| B | 59.66 | 45 | 45.66 | 73 | 48 |

5. Discussion

The clusters of the vehicle crashes were found based on similarity at both streets. Using the NCSS software, the Analysis of Variance (ANOVA) for the K-means algorithm is shown in **Table 4**. We can see that for street A, the mean square Between is 395.4167, the mean square Within is 39.7619, the F-ratio is 9.94 and the p-value is 0.00515. While for street B, the mean square Between is 315.1667, the mean square Within is 56.4761, the F-ratio is 5.58 and the p-value is 0.00436. The p-value is very small and significant at both streets indicating perfect model fit for the K-means algorithm.

Table 4. The ANOVA table of vehicle crashes at street A and street B.

| Variables | Degrees of Freedom | | Mean Square | | | |
|----------------------------|--------------------|-----|-------------|---------|---------|---------|
| | DF1 | DF2 | Between | Within | F-ratio | p-value |
| Number of crashes street A | 4 | 7 | 395.4167 | 39.7619 | 9.94 | 0.00515 |
| Number of crashes street B | 4 | 7 | 315.1667 | 56.4761 | 5.58 | 0.00436 |

The distance matrix of each cluster center using the Euclidian distances is

shown in **Table 5**. These distances play a crucial role in creating the crash clusters. We can see for example that cluster 2 consists of 2 crash points, the first crash point is at (distance 2.0071 from cluster 1, distance 1.4364 from cluster 3, distance 2.9558 from cluster 4, and distance 3.2454 from cluster 5). The second crash point is at (distance 2.3071 from cluster 1, distance 1.8640 from cluster 3, distance 3.7024 from cluster 4 and distance 3.4701 from cluster 5).

Table 5. The distance matrix of each cluster center by Euclidian distances.

| Cluster | 1 | 2 | 3 | 4 | 5 |
|---------|--------|--------|--------|--------|--------|
| 2 | 2.0071 | 0.6628 | 1.4364 | 2.9558 | 3.2454 |
| 2 | 2.3071 | 0.6628 | 1.8640 | 3.7024 | 3.4701 |
| 3 | 2.5605 | 1.3936 | 0.4109 | 3.1438 | 2.3642 |
| 3 | 1.7887 | 1.2536 | 0.6155 | 2.3569 | 2.2151 |
| 3 | 2.6933 | 2.0870 | 0.5767 | 2.8674 | 1.6997 |
| 1 | 1.0275 | 2.5548 | 2.9625 | 1.7938 | 3.5321 |
| 1 | 0.8641 | 1.6359 | 1.5525 | 1.9589 | 2.1455 |
| 4 | 1.6791 | 3.2524 | 3.0463 | 0.7675 | 3.0666 |
| 1 | 0.6435 | 2.3921 | 2.6757 | 2.1034 | 2.8611 |
| 5 | 2.6200 | 3.0594 | 1.7779 | 2.2558 | 0.2916 |
| 4 | 2.1293 | 3.4879 | 2.6618 | 0.7675 | 1.7562 |
| 5 | 2.9447 | 3.5363 | 2.3121 | 2.5282 | 0.2916 |

The matrix of cluster membership probability for all crash points in Fuzzy C-means algorithm is shown in **Table 6**. We can see for example that cluster 2 has 2 crash points, the first point has (probability of 0.1305 in cluster 1, probability of 0.6258 in cluster 2, probability of 0.0767 in cluster 3, probability of 0.0887 in cluster 4, and probability of 0.0783 in cluster 5). While the second point has (probability of 0.1072 in cluster 1, probability of 0.6543 in cluster 2, probability of 0.0681 in cluster 3, probability of 0.0772 in cluster 4, and probability of 0.0932 in cluster 5). The membership probability of each crash point is vital in assigning the crash points to different clusters in fuzzy C-means algorithm. This shows that each crash point can be assigned to more than one cluster as determined by its membership probability.

Goodness of fits statistics for the fuzzy C-means are shown in **Table 7**. We need to choose the optimal number of clusters (C) in Fuzzy C-means so that $F_c(U)$ is large, and $D_c(U)$ is small. We can see that $F_c(U)$ has the largest value when using 5 clusters (0.5505), and $D_c(U)$ has the smallest value when using 5 clusters (0.1087) as shown in **Table 7**. Therefore, we will keep our initial C value of 5 clusters. The goodness of fits is essential in obtaining the optimal C value for the fuzzy algorithm.

Table 6. The matrix of cluster membership probability for all crash points.

| Cluster ID | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 | Cluster 5 |
|------------|-----------|-----------|-----------|-----------|-----------|
| 1 | 0.4859 | 0.1969 | 0.1437 | 0.0973 | 0.0763 |
| 1 | 0.8532 | 0.0543 | 0.0415 | 0.0269 | 0.0241 |
| 1 | 0.0938 | 0.7504 | 0.0487 | 0.0514 | 0.0557 |
| 2 | 0.1305 | 0.6258 | 0.0767 | 0.0887 | 0.0783 |
| 2 | 0.1072 | 0.6543 | 0.0681 | 0.0772 | 0.0932 |
| 3 | 0.0768 | 0.0588 | 0.7690 | 0.0573 | 0.0382 |
| 3 | 0.1949 | 0.1394 | 0.4889 | 0.0978 | 0.0790 |
| 4 | 0.0115 | 0.0143 | 0.0120 | 0.9481 | 0.0141 |
| 3 | 0.1307 | 0.0955 | 0.6287 | 0.0823 | 0.0627 |
| 5 | 0.0310 | 0.0492 | 0.0259 | 0.0458 | 0.8482 |
| 4 | 0.1112 | 0.1611 | 0.1116 | 0.3967 | 0.2194 |
| 5 | 0.0296 | 0.0446 | 0.0256 | 0.0429 | 0.8574 |

Table 7. Goodness of Fits for the fuzzy C-means algorithm.

| Number of Clusters | Average | | Goodness-of-Fit Statistics | | | |
|--------------------|----------|------------|----------------------------|----------|--------|----------|
| | Distance | Silhouette | $F(U)$ | $F_c(U)$ | $D(U)$ | $D_c(U)$ |
| 2 | 2.951851 | 0.334028 | 0.6048 | 0.2095 | 0.1988 | 0.3975 |
| 3 | 1.676369 | 0.511612 | 0.6079 | 0.4119 | 0.1815 | 0.2722 |
| 4 | 1.097818 | 0.572472 | 0.6047 | 0.4729 | 0.1582 | 0.2109 |
| 5 | 0.698199 | 0.702357 | 0.6404 | 0.5505 | 0.0870 | 0.1087 |

6. Conclusion

Clustering is a powerful unsupervised machine learning algorithm for grouping unlabeled datasets. In order to show different patterns in vehicle crashes, this paper presented two clustering algorithms, the K-means and the Fuzzy C-means. In both algorithms, the goal is to identify cluster centers that best represent the data points in each cluster. The cluster centers are calculated as the mean or centroid of the cluster's assigned data points. The algorithms iteratively update the cluster centers until convergence to find the optimal clusters. The C-means clustering can assign a probability score for each data point for being in that cluster. Therefore, fuzzy C-means clustering gives better results for overlapped data sets compared to K-means clustering.

Conflicts of Interest

The author declares no conflicts of interest.

References

- [1] Hastie, T., Tibshirani, R. and Friedman, J. (2008) The Elements of Statistical Learn-

- ing. Springer.
- [2] Lawless, J.F. (2002) Statistical Models and Methods for Lifetime Data. John Wiley & Sons, Inc. <https://doi.org/10.1002/9781118033005>
 - [3] Abdulhafedh, A. (2021) Incorporating K-Means, Hierarchical Clustering and PCA in Customer Segmentation. *Journal of City and Development*, **3**, 12-30.
 - [4] Imbens, G.W. and Rubin, D.B. (2015) Causal Inference for Statistics, Social, and Biomedical Sciences. Cambridge University Press. <https://doi.org/10.1017/cbo9781139025751>
 - [5] Waggoner, P.D. (2020) Unsupervised Machine Learning for Clustering in Political and Social Research. Cambridge University Press. <https://doi.org/10.1017/9781108883955>
 - [6] Bradford Tuckfield (2019) Applied Unsupervised Learning with R: Uncover Hidden Relationships and Patterns with K-Means Clustering, Hierarchical Clustering, and PCA. Packt Publishing.
 - [7] Colins, M. (2017) Machine Learning: An Introduction to Supervised and Un-Supervised Learning Algorithms. CreateSpace.
 - [8] Celebi, M.E. and Aydin, K. (2016) Unsupervised Learning Algorithms. Springer.
 - [9] Ever-Hadani, S. (1980) Applications of Cluster Analysis Algorithm to Geostatistical Series. *Regional Science and Urban Economics*, **10**, 123-151. [https://doi.org/10.1016/0166-0462\(80\)90052-6](https://doi.org/10.1016/0166-0462(80)90052-6)
 - [10] Ghosh, S. and Dubey, S.K. (2013) Comparative Analysis of K-Means and Fuzzy C-Means Algorithms. *International Journal of Advanced Computer Science and Applications*, **4**, 35-39. <https://doi.org/10.14569/ijacsa.2013.040406>
 - [11] Hamerly, G. and Elkan, C. (2002) Alternatives to the K-Means Algorithm that Find Better Clusterings. *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, New York, 4-9 November 2002, 600-607. <https://doi.org/10.1145/584792.584890>
 - [12] Bezdek, J.C. (1981) Pattern Recognition with Fuzzy Objective Function Algorithms. Springer.
 - [13] Bradley, P.S. and Fayyad, U.M. (1998) Refining Initial Points for K-Means Clustering. *Proceedings of the 15th International Conference on Machine Learning*, Madison, 24-27 July 1998, 91-99.
 - [14] Kalton, A., Langley, P., Wagstaff, K. and Yoo, J. (2001) Generalized Clustering, Supervised Learning, and Data Assignment. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, 26-29 August 2001, 299-304. <https://doi.org/10.1145/502512.502555>
 - [15] Kearns, M., Mansour, Y. and Ng, A.Y. (1997) An Information-Theoretic Analysis of Hard and Soft Assignment Methods for Clustering. *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI1997)*, Providence, 1-3 August 1997, 282-293.
 - [16] Pelleg, D. and Moore, A. (2000) X-Means: Extending K-Means with Efficient Estimation of the Number of Clusters. *Proceedings of the 17th International Conference on Machine Learning (ICML2000)*, Stanford, 29 June-2 July 2000, 727-734.
 - [17] Cannon, R.L., Dave, J.V. and Bezdek, J.C. (1986) Efficient Implementation of the Fuzzy C-Means Clustering Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**, 248-255. <https://doi.org/10.1109/tpami.1986.4767778>
 - [18] Cebeci, Z. and Yildiz, F. (2015) Comparison of K-Means and Fuzzy C-Means Algo-

- rithms on Different Cluster Structures. *Journal of Agricultural Informatics*, **6**, 13-23. <https://doi.org/10.17700/jai.2015.6.3.196>
- [19] Cai, W., Chen, S. and Zhang, D. (2007) Fast and Robust Fuzzy C-Means Clustering Algorithms Incorporating Local Information for Image Segmentation. *Pattern Recognition*, **40**, 825-838. <https://doi.org/10.1016/j.patcog.2006.07.011>
- [20] Dhanachandra, N., Manglem, K. and Chanu, Y.J. (2015) Image Segmentation Using K-Means Clustering Algorithm and Subtractive Clustering Algorithm. *Procedia Computer Science*, **54**, 764-771. <https://doi.org/10.1016/j.procs.2015.06.090>
- [21] Gath, I. and Geva, A.B. (1989) Unsupervised Optimal Fuzzy Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**, 773-780. <https://doi.org/10.1109/34.192473>
- [22] Kim, D., Lee, K.H. and Lee, D. (2004) On Cluster Validity Index for Estimation of the Optimal Number of Fuzzy Clusters. *Pattern Recognition*, **37**, 2009-2025. <https://doi.org/10.1016/j.patcog.2004.04.007>
- [23] Liu, Y., Li, Z., Xiong, H., Gao, X. and Wu, J. (2010) Understanding of Internal Clustering Validation Measures. 2010 *IEEE International Conference on Data Mining*, Sydney, 13-17 December 2010, 911-916. <https://doi.org/10.1109/icdm.2010.35>
- [24] Ng, H.P., Ong, S.H., Foong, K.W.C., Goh, P.S. and Nowinski, W.L. Medical Image Segmentation Using K-Means Clustering and Improved Watershed Algorithm. 2006 *IEEE Southwest Symposium on Image Analysis and Interpretation*, Denver, 26-28 March 2006, 61-65. <https://doi.org/10.1109/ssiai.2006.1633722>
- [25] Wu, K. (2012) Analysis of Parameter Selections for Fuzzy C-Means. *Pattern Recognition*, **45**, 407-415. <https://doi.org/10.1016/j.patcog.2011.07.012>
- [26] Xie, X.L. and Beni, G. (1991) A Validity Measure for Fuzzy Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 841-847. <https://doi.org/10.1109/34.85677>
- [27] Hamerly, G.J. (2003) Learning Structure and Concepts in Data through Data Clustering. University of California.
- [28] Ruspini, E.H., Bezdek, J.C. and Keller, J.M. (2019) Fuzzy Clustering: A Historical Perspective. *IEEE Computational Intelligence Magazine*, **14**, 45-55. <https://doi.org/10.1109/mci.2018.2881643>
- [29] Dunn, J.C. (1974) Well-Separated Clusters and Optimal Fuzzy Partitions. *Journal of Cybernetics*, **4**, 95-104. <https://doi.org/10.1080/01969727408546059>
- [30] Kaufman, L. and Rousseeuw, P.J. (1990) Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, Inc. <https://doi.org/10.1002/9780470316801>