Grid-Based Approach to Determining Parameters of the DBSCAN Algorithm

Artur Starczewski[1(✉)] and Andrzej Cader[2,3]

[1] Institute of Computational Intelligence, Częstochowa University of Technology, Al. Armii Krajowej 36, 42-200 Częstochowa, Poland
artur.starczewski@iisi.pcz.pl
[2] Information Technology Institute, University of Social Sciences, 90-113 Łódź, Poland
acader@san.edu.pl
[3] Clark University, Worcester, MA 01610, USA

Abstract. Clustering is a very important technique used in many fields in order to deal with large datasets. In clustering algorithms, one of the most popular approaches is based on an analysis of clusters density. Density-based algorithms include different methods but the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is one of the most cited in the scientific literature. This algorithm can identify clusters of arbitrary shapes and sizes that occur in a dataset. Thus, the DBSCAN is very widely applied in various applications and has many modifications. However, there is a key issue of the right choice of its two input parameters, i.e the neighborhood radius (*eps*) and the *MinPts*. In this paper, a new method for determining the neighborhood radius (*eps*) and the *MinPts* is proposed. This method is based on finding a proper grid of cells for a dataset. Next, the grid is used to calculate the right values of these two parameters. Experimental results have been obtained for several different datasets and they confirm a very good performance of the newly proposed method.

Keywords: Clustering algorithms · Data mining · DBSCAN

1 Introduction

Clustering algorithms discover naturally occurring structures in datasets. Nowadays, extensive collections of data pose a great challenge for clustering algorithms. So, many researchers create different new clustering algorithms or modify existing approaches [5,6,11,19,21]. It is worth noting that data clustering is applied in various areas, e.g. biology, spatial data analysis, or business. The key issue is the right choice of input parameters because the same algorithm can produce different results depending on applied parameters. This problem can be resolved by using different cluster validity indices, e.g., [10,26,29,30]. Generally, clustering algorithms can be divided into four categories: partitioning,

© Springer Nature Switzerland AG 2020
L. Rutkowski et al. (Eds.): ICAISC 2020, LNAI 12415, pp. 555–565, 2020.
https://doi.org/10.1007/978-3-030-61401-0_52

hierarchical, grid-based, and density-based clustering. Well-known partitioning algorithms include the K-means or Partitioning Around Medoids (PAM) [3,32]. The next clustering category called hierarchical is based on an agglomerative or divisive approach, e.g. the Single-linkage, Complete-linkage, Average-linkage, or Divisive ANAlysis Clustering (DIANA) [17,23]. On the other hand, the grid-based approach creates a grid of cells for a dataset, e.g. the Statistical Information Grid-based (STING) or Wavelet-based Clustering (WaveCluster) methods [18,28,31]. The last category can be represented by the Density-Based Spatial Clustering of Application with Noise (DBSCAN) algorithm [9] which has many modifications [7,12,13,15,27]. This algorithm can discover clusters of an arbitrary shape and size but requires two input parameters, i.e. the *eps* and the $MinPts$. The determination of these parameters is very important for the DBSCAN algorithm to work properly. It is important to note that clustering methods can be used during the process of designing various neural networks [1,2], fuzzy, and rule systems [4,8,14,16,20,22,24,25].

In this paper, a new approach to determining the *eps* and $MinPts$ parameters is proposed. It is based on the creation of a proper grid of cells and the grid is used to define the values of the two parameters. This paper is organized as follows: Sect. 2 presents a description of the $DBSCAN$ clustering algorithm. In Sect. 3 the new method for determining the parameters is outlined, while Sect. 4 illustrates the experimental results on datasets. Finally, Sect. 5 presents the conclusions.

2 The DBSCAN Algorithm

The concept of the $DBSCAN$ algorithm is presented in this section. As mentioned above, this algorithm is very popular, because it can find clusters of arbitrary shapes and requires only two input parameters, i.e. the *eps* and the $MinPts$. The *eps* is usually determined by the user and it has a large influence on the creation of clusters. The next parameter, i.e. the $MinPts$ is the minimal number of neighboring points belonging to the so-called *core point*. Let us denote a dataset by X, where point $p \in X$. The following definitions (see [9]) will be helpful in understanding how the DBSCAN algorithm works.

Definition 1: The *eps-neighborhood* of point $p \in X$ is called $N_{eps}(p)$ and is defined as follows: $N_{eps}(p) = \{q \in X | dist(p,q) \leq eps\}$, where $dist(p,q)$ is a distance function between p and q.

Definition 2: p is called the *core* if the number of points belonging to $N_{eps}(p)$ is greater or equal to the $MinPts$.

Definition 3: Point q is *directly density-reachable* from point p (for the given *eps* and the $MinPts$) if p is the *core point* and q belongs to $N_{eps}(p)$.

Definition 4: if point q is *directly density-reachable* from point p and the number of points belonging to $N_{eps}(q)$ is smaller than the $MinPts$, q is called a *border point*.

Definition 5: Point q is a *noise* if it is neither a *core point* nor a *border point*.

Definition 6: Point q is *density-reachable* from point p (for the given *eps* and the $MinPts$) if there is a chain of points $q_1, q_2, ..., q_n$ and $q_1 = p$, $q_n = q$, so that q_{i+1} is *directly density-reachable* from q_i

Definition 7: Point q is *density-connected* to point p (for the given *eps* and the $MinPts$) if there is point o such that q and p are *density-reachable* from point o.

Definition 8: Cluster C (for the given *eps* and the $MinPts$) is a non-empty subset of X and the following conditions are satisfied: first, $\forall p, q$: if $p \in C$ and q is *density-reachable* from p, then $q \in C$, next $\forall p, q \in C$: p is *density-connected* to q.

The DBSCAN algorithm creates clusters according to the following: at first, point p is selected randomly if $|N_{eps}(p)| \geq MinPts$, than point p will be the *core point* and a new cluster will be created. Next, the new cluster is expanded by the points which are *density-reachable* from p. This process is repeated until no cluster is found. On the other hand, if $|N_{eps}(p)| < MinPts$, then point p will be a *noise*, but this point can be included in another cluster if it is *density-reachable* from some *core point*.

3 Grid-Based Approach to Determining the Eps and MinPts Parameters

The right choice of the *eps* and *MinPts* parameters is a fundamental issue for the high performance of the DBSCAN algorithm. The proposed method is based on a uniform grid of cells which is created for a dataset. In order to provide a clearer explanation of this new approach, an example of a 2-dimensional dataset is generated. Figure 1 shows this dataset consisting of 1200 elements located in four clusters, i.e. 200, 250, 300 and 450 elements per cluster, respectively. Next, for this dataset an example grid of cells can be created, e.g. consisting of 100 cells (10 x 10). Figure 2 shows this uniform grid of cells. It can be noted that the

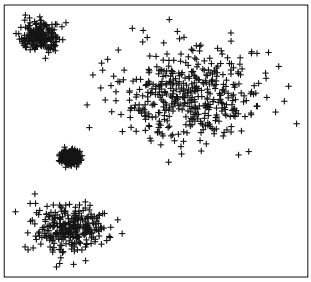

Fig. 1. An example of a 2-dimensional dataset consisting of four clusters.

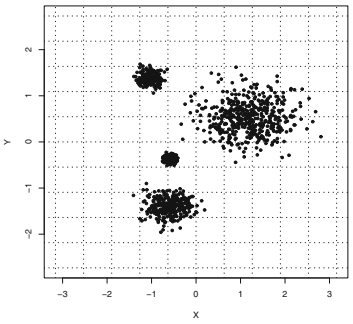

Fig. 2. Uniform grid consisting of 100 cells (10 x 10) for the example dataset.

proper grid can be used to define the value of the *eps* parameter, but the key issue is an appropriate choice of the size of the grid, which has a big influence on the value of the parameter. In this new method, a way of solving this problem is proposed and it consists of a few steps. First, several grids of cells are created, where the size of rows and columns of grids change in a wide range, i.e. from 2 to 90 (2 x 2 and 90 x 90 cells). So, the number of cells is changed from 4 to 8100. Such a number of cells gives precise information about the properties of a dataset. Let us denote the size of a grid by G_{size}. For all the created grids, three ranges can be defined as in the following:

$$\begin{array}{llll} range1 & for & 2 \leq & G_{size} \leq 30 \\ range2 & for & 30 < & G_{size} \leq 60 \\ range3 & for & 60 < & G_{size} \leq 90 \end{array} \tag{1}$$

It is worth noting that the second parameter of the DBSCAN algorithm, i.e. the *MinPts* is also very important and it affects a number of so-called noise data. Generally, the choice of this parameter is often realized individually depending on a dataset, but very often the $MinPts$ equals 4, 5, or 6. Such values of this parameter ensure a good compromise between the size of clusters and an amount of noise data in most cases. So, in this new approach, the values of the $MinPts$ are selected from 4 to 6. As mentioned above, the sizes of the grids range from 2 to 90. Next, in all the created grids are found cells which include only 4 elements. Then, the grid which includes a maximum number of cells with four elements is found and the size of the grid is noted by G_{max4}. Furthermore, the grids which include a maximum number of cells with 5 and 6 elements are also found and the sizes of grids are noted by G_{max5} and G_{max6}. In the next step, the $dist_4$, $dist_5$ and $dist_6$ parameters are determined for the G_{max4}, G_{max5} and G_{max6} grid sizes, respectively. The values of these parameters are maximum distances between the elements of the cells which include 4, 5, and 6 elements, respectively. Next, if condition ($G_{max4} > G_{max5} > G_{max6}$) is fulfilled, the value of the *eps* is defined as follows:

$$eps = \begin{cases} a * dist_4 & for \quad G_{max4} \in range1 \\ b * dist_5 & for \quad G_{max4} \in range2 \\ c * dist_6 & for \quad G_{max4} \in range3 \end{cases} \tag{2}$$

where factors a, b and c are experimentally determined and their values are 1, 1.2 and 1.5, respectively. On the other hand, the $MinPts$ is expressed as follows:

$$MinPts = \begin{cases} 4 & for \quad G_{max4} \in range1 \\ 5 & for \quad G_{max4} \in range2 \\ 6 & for \quad G_{max4} \in range3 \end{cases} \tag{3}$$

Sometimes, for different datasets condition ($G_{max4} > G_{max5} > G_{max6}$) may not be fulfilled. This means that the clusters have a different density because when the $MinPts$ increases and the clusters have a similar density, the maximum number of cells should be decreased. In these cases, when the condition is not fulfilled the values of the a, b, and c factors should be increased so that they equal 2. In Table 1 are presented the values of G_{max4}, G_{max5} and G_{max6} calculated for the example dataset. It can be observed that for the $MinPts$ equal to 5, the size of the grid is larger than the size for the $MinPts$ equal to 4. So, clusters are of different density in the dataset (see Fig. 1). Condition ($G_{max4} > G_{max5} > G_{max6}$) is not fulfilled and the b parameter is increased (equals 2). Moreover, when the $MinPts$ is equal to 4, G_{max4} is 52 and is included in $range2$. Thus, $eps = b * dist_5$ (see Eq. 2) and the values of the eps and $MinPts$ parameters are 0.20 and 5, respectively. Such values of input parameters are used in the $DBSCAN$ algorithm.

Table 1. Values of G_{max4}, G_{max5} *and* G_{max6} for the example dataset

Maximum number of cells	Values of the $MinPts$	Number of cells
$G_{max4} = 52$ (52 x 52−2704 cells)	4	33
$G_{max5} = 62$ (62 x 62−3844 cells)	5	24
$G_{max6} = 48$ (48 x 48−864 cells)	6	17

Figure 3 shows the results of the $DBSCAN$ clustering algorithm for the example dataset. In the next section, the results of the experimental tests are presented to confirm the effectiveness of the new approach.

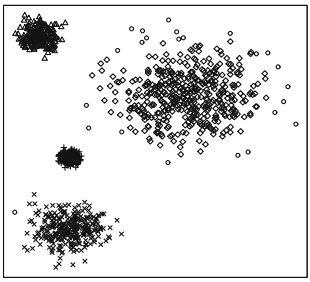

Fig. 3. Results of the *DBSCAN* clustering algorithm for the example dataset.

4 Experimental Results

In this section, several experiments have been conducted on 2-dimensional artificial datasets. In these experiments, the *DBSCAN* algorithm is used to cluster the data. As mentioned above, the *eps* and *MinPts* parameters play a very important role in creating correct clusters by this clustering algorithm. So, they are defined based on the new method described in Sect. 3 and the calculated values of these parameters are presented in Table 3. Moreover, the evaluation of the accuracy of the DBSCAN algorithm is conducted by a visual inspection. It is worth noting that the artificial datasets include clusters of various shapes and sizes. On the other hand, for clustering multidimensional datasets, determining the input parameters of the DBSCAN algorithm is very difficult.

Table 2. A detailed description of the artificial datasets

Datasets	No. of elements	Clusters
Data 1	700	2
Data 2	700	3
Data 3	3000	3
Data 4	1000	3
Data 5	900	4
Data 6	500	4
Data 7	500	4
Data 8	1800	5
Data 9	700	6

Table 3. The *eps* and $MinPts$ values used by the DBSCAN algorithm

Datasets	*eps*	$MinPts$
Data 1	0.36	5
Data 2	0.22	4
Data 3	0.16	6
Data 4	0.20	5
Data 5	0.34	6
Data 6	0.33	5
Data 7	0.20	5
Data 8	0.23	5
Data 9	0.21	5

4.1 Datasets

In the conducted experiments nine 2-dimensional datasets are used. Most of them come from the *R* package. The artificial data are called *Data* 1, *Data* 2, *Data* 3, *Data* 4, *Data* 5, *Data* 6, *Data* 7, *Data* 8 and *Data* 9, respectively. They consist of a various number of clusters, i.e. 2, 3, 4, 5, and 6 clusters. The scatter plot of these data is presented in Fig. 4. As it can be observed on the plot, the clusters are located in different areas and some of the clusters are very close to each other and the others are quite far apart. For instance, *Data* 1 is a so-called spirals problem, where the points are on two entangled spirals, in *Data* 5 the elements create a Gaussian, square, triangle and wave shapes and *Data* 6 consists of 2 Gaussian eyes, a trapezoid nose and a parabola mouth (with a vertical Gaussian one). Moreover, the sizes of the clusters are different and they contain a various number of elements. In Table 2 is shown a description of these datasets.

4.2 Experiments

The experimental analysis is designed to evaluate the performance of the new method to specify the *eps* and $MinPts$ parameters. As mentioned above, these parameters are very important for the $DBSCAN$ algorithm to work correctly. In standard approaches, they are determined by a visual inspection of the sorted values of a function which computes a distance between each element of a dataset and its k-th nearest neighbor. The new approach described in Sect. 3 is based on finding a proper grid of cells and it makes it possible to determine these two input parameters. In these experiments, the nine 2-dimensional datasets used are called *Data* 1, *Data* 2, *Data* 3, *Data* 4, *Data* 5, *Data* 6, *Data* 7, *Data* 8 and *Data* 9 datasets. It is worth noting that the value of the $MinPts$ parameter is also chosen when the size of the grid changes from 2 to 90 (2 x 2 and 90 x 90 cells). Then, when these parameters are specified by the new method, the $DBSCAN$ algorithm can be used to cluster these datasets. Figure 5 shows the results of

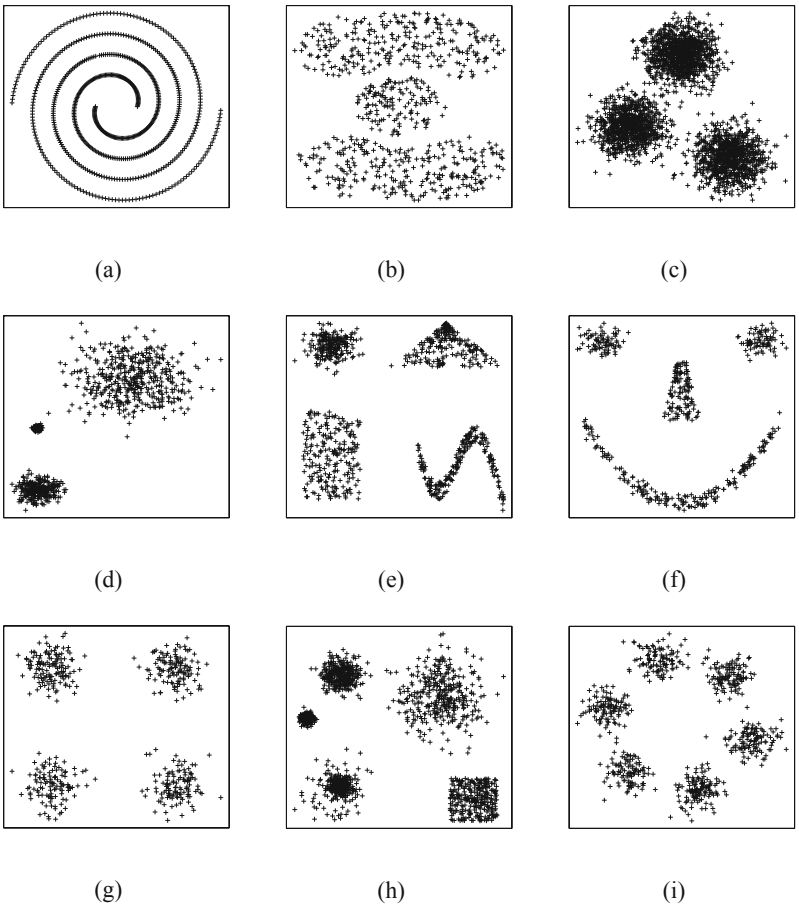

Fig. 4. Examples of 2-dimensional artificial datasets: (a) *Data* 1, (b) *Data* 2, (c) *Data* 3, (d) *Data* 4, (e) *Data* 5, (f) *Data* 6, (g) *Data* 7, (h) *Data* 8 and (i) *Data* 9.

the *DBSCAN* algorithm, where each cluster is marked with different signs. The data elements classified as the *noise* are marked with a circle. Thus, despite the fact that the differences in the distances and the shapes between clusters are significant, all the datasets are clustered correctly by the $DBSCAN$. Moreover, a number of the data elements classified as noise in all the datasets is relatively insignificant.

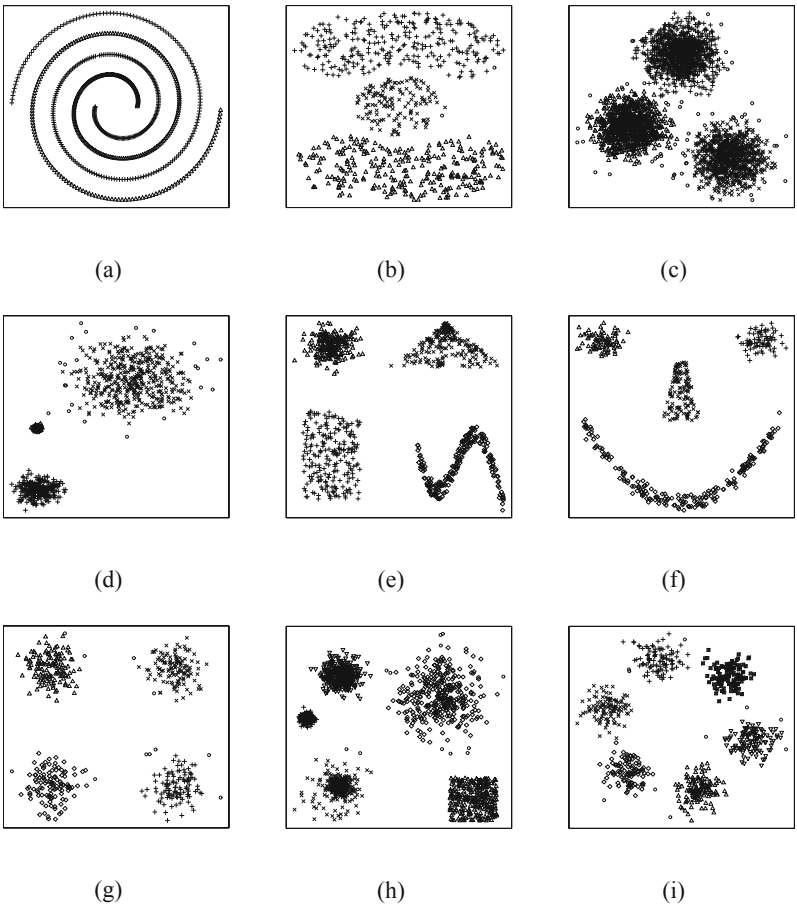

Fig. 5. Results of the *DBSCAN* clustering algorithm for 2-dimensional datasets: (a) *Data* 1, (b) *Data* 2, (c) *Data* 3, (d) *Data* 4, (e) *Data* 5, (f) *Data* 6, (g) *Data* 7, (h) *Data* 8 and (i) *Data* 9

5 Conclusions

In this paper, a new approach is proposed to calculate the *eps* and $MinPts$ parameters of the DBSCAN algorithm. It is based on finding the right grid of cells, which is selected from many other grids. As mentioned above, the sizes of the grids change from 2 to 90. It is worth noting that the determination of the $MinPts$ parameter is also difficult and it is often chosen empirically depending on datasets being investigated. In this new method, the values of the $MinPts$ parameter are selected from 4 to 6. Generally, the right grid of cells makes it possible to correctly calculate these two input parameters. In the conducted experiments, several 2-dimensional datasets were used, where a

number of clusters, sizes, and shapes were very different. All the presented results confirm the high efficiency of the newly proposed approach.

References

1. Bilski, J., Smoląg, J., Żurada, J.M.: Parallel approach to the Levenberg-Marquardt learning algorithm for feedforward neural networks. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2015. LNCS (LNAI), vol. 9119, pp. 3–14. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19324-3_1
2. Bilski, J., Wilamowski, B.M.: Parallel Levenberg-Marquardt algorithm without error backpropagation. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2017. LNCS (LNAI), vol. 10245, pp. 25–39. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59063-9_3
3. Bradley, P., Fayyad, U.: Refining initial points for k-means clustering. In: Proceedings of the Fifteenth International Conference on Knowledge Discovery and Data Mining, pp. 9–15. AAAI Press, New York (1998)
4. Bologna, G., Hayashi, Y.: Characterization of symbolic rules embedded in deep DIMLP networks: a challenge to transparency of deep learning. J. Artif. Intell. Soft Comput. Res. **7**(4), 265–286 (2017)
5. Chen, X., Liu, W., Qui, H., Lai, J.: APSCAN: a parameter free algorithm for clustering. Pattern Recogn. Lett. **32**, 973–986 (2011)
6. Chen, J.: Hybrid clustering algorithm based on PSO with the multidimensional asynchronism and stochastic disturbance method. J. Theor. Appl. Inform. Technol. **46**, 343–440 (2012)
7. Chen, Y., Tang, S., Bouguila, N., Wang, C., Du, J., Li, H.: A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data. Pattern Recogn. **83**, 375–387 (2018)
8. D'Aniello, G., Gaeta, M., Loia, F., Reformat, M., Toti, D.: An environment for collective perception based on fuzzy and semantic approaches. J. Artif. Intell. Soft Comput. Res. **8**(3), 191–210 (2018)
9. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceeding of 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)
10. Fränti, P., Rezaei, M., Zhao, Q.: Centroid index: cluster level similarity measure. Pattern Recogn. **47**(9), 3034–3045 (2014)
11. Hruschka, E.R., de Castro, L.N., Campello, R.J.: Evolutionary algorithms for clustering gene-expression data. In: Data Mining, Fourth IEEE International Conference on Data Mining (ICDM 2004), pp. 403–406. IEEE (2004)
12. Karami, A., Johansson, R.: Choosing DBSCAN parameters automatically using differential evolution. Int. J. Comput. Appl. **91**, 1–11 (2014)
13. Lai, W., Zhou, M., Hu, F., Bian, K., Song, Q.: A new DBSCAN parameters determination method based on improved MVO. IEEE Access **7**, 104085–104095 (2019)
14. Liu, H., Gegov, A., Cocea, M.: Rule based networks: an efficient and interpretable representation of computational models. J. Artif. Intell. Soft Comput. Res. **7**(2), 111–123 (2017)
15. Luchi, D., Rodrigues, A.L., Varejao, F.M.: Sampling approaches for applying DBSCAN to large datasets. Pattern Recogn. Lett. **117**, 90–96 (2019)

16. Ferdaus, M.M., Anavatti, S.G., Matthew, A., Pratama, G., Pratama, M.: Development of C-means clustering based adaptive fuzzy controller for a flapping wing micro air vehicle. J. Artif. Intell. Soft Comput. Res. **9**(2), 99–109 (2019). https://doi.org/10.2478/jaiscr-2018-0027
17. Murtagh, F.: A survey of recent advances in hierarchical clustering algorithms. Comput. J. **26**(4), 354–359 (1983)
18. Patrikainen, A., Meila, M.: Comparing subspace clusterings. IEEE Trans. Knowl. Data Eng. **18**(7), 902–916 (2006)
19. Pei, Z., Hua, X., Han, J.: The clustering algorithm based on particle swarm optimization algorithm. In: Proceedings of the 2008 International Conference on Intelligent Computation Technology and Automation, Washington, USA, vol. 1, pp. 148–151 (2008)
20. Prasad, M., Liu, Y.-T., Li, D.-L., Lin, C.-T., Shah, R.R., Kaiwartya, O.P.: A new mechanism for data visualization with TSK-type preprocessed collaborative fuzzy rule based system. J. Artif. Intell. Soft Comput. Res. **7**(1), 33–46 (2017)
21. Rastin, P., Matei, B., Cabanes, G., Grozavu, N., Bennani, Y.: Impact of learners' quality and diversity in collaborative clustering. J. Artif. Intell. Soft Comput. Res. **9**(2), 149–165 (2019). https://doi.org/10.2478/jaiscr-2018-0030
22. Riid, A., Preden, J.-S.: Design of fuzzy rule-based classifiers through granulation and consolidation. J. Artif. Intell. Soft Comput. Res. **7**(2), 137–147 (2017)
23. Rohlf, F.: Single-link clustering algorithms. In: Krishnaiah, P.R., Kanal, L.N., (eds.) Handbook of Statistics, vol. 2, pp. 267–284 (1982)
24. Rutkowski, T., Łapa, K., Nielek, R.: On explainable fuzzy recommenders and their performance evaluation. Int. J. Appl. Math. Comput. Sci. **29**(3), 595–610 (2019). https://doi.org/10.2478/amcs-2019-0044
25. Rutkowski, T., Łapa, K., Jaworski, M., Nielek, R., Rutkowska, D.: On explainable flexible fuzzy recommender and its performance evaluation using the akaike information criterion. In: Gedeon, T., Wong, K.W., Lee, M. (eds.) ICONIP 2019. CCIS, vol. 1142, pp. 717–724. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36808-1_78
26. Sameh, A.S., Asoke, K.N.: Development of assessment criteria for clustering algorithms. Pattern Anal. Appl. **12**(1), 79–98 (2009)
27. Shah G.H.: An improved DBSCAN, a density based clustering algorithm with parameter selection for high dimensional data sets. In: Nirma University International Engineering (NUiCONE), pp. 1–6 (2012)
28. Sheikholeslam, G., Chatterjee, S., Zhang, A.: WaveCluster: a wavelet-based clustering approach for spatial data in very large databases. Int. J. Very Large Data Bases **8**(3–4), 289–304 (2000)
29. Shieh, H.-L.: Robust validity index for a modified subtractive clustering algorithm. Appl. Soft Comput. **22**, 47–59 (2014)
30. Starczewski, A.: A new validity index for crisp clusters. Pattern Anal. Appl. **20**(3), 687–700 (2017)
31. Wang, W., Yang, J., Muntz, R.: STING: a statistical information grid approach to spatial data mining. In: Proceedings of the 23rd International Conference on Very Large Data Bases. (VLDB 1997), pp. 186–195 (1997)
32. Zalik, K.R.: An efficient k-means clustering algorithm. Pattern Recogn. Lett. **29**(9), 1385–1391 (2008)