Scientific Research Publishing

Open Journal of Applied Sciences, 2023, 13, 1163-1177
https://www.scirp.org/journal/ojapps
ISSN Online: 2165-3925
ISSN Print: 2165-3917

Study on the Development and Implementation of Different Big Data Clustering Methods

Jean Pierre Ntayagabiri[1], Jérémie Ndikumagenge[1], Longin Ndayisaba[1], Boribo Kikunda Philippe[2]

[1]Center of Research in Infrastructure, Environment and Technology (CRIET), University of Burundi, Bujumbura, Burundi
[2]Catholic University of Bukavu, Bukavu, Democratic Republic of the Congo
Email: jpntayaga2@gmail.com, jeremie.ndikumagenge@ub.edu.bi, longin.ndayisaba@ub.edu.bi, boribokikunda@gmail.com

How to cite this paper: Ntayagabiri, J.P., Ndikumagenge, J., Ndayisaba, L. and Philippe, B.K. (2023) Study on the Development and Implementation of Different Big Data Clustering Methods. *Open Journal of Applied Sciences*, **13**, 1163-1177.
https://doi.org/10.4236/ojapps.2023.137092

Received: October 28, 2022
Accepted: July 25, 2023
Published: July 28, 2023

Abstract

Clustering is an unsupervised learning method used to organize raw data in such a way that those with the same (similar) characteristics are found in the same class and those that are dissimilar are found in different classes. In this day and age, the very rapid increase in the amount of data being produced brings new challenges in the analysis and storage of this data. Recently, there is a growing interest in key areas such as real-time data mining, which reveal an urgent need to process very large data under strict performance constraints. The objective of this paper is to survey four algorithms including K-Means algorithm, FCM algorithm, EM algorithm and BIRCH, used for data clustering and then show their strengths and weaknesses. Another task is to compare the results obtained by applying each of these algorithms to the same data and to give a conclusion based on these results.

Keywords

Clustering, K-Means, Fuzzy c-Means, Expectation Maximization, BIRCH

1. Introduction

In recent years, we have entered a new era, that of data. Ever since humans have been able to write or even communicate, they have been collecting and transmitting data. According to the International Data Corporation (IDC), it is known approximately that from the first traces of man until 2005, the human species created 135 exabytes of data and 161 exabytes of data were created until 2006 [1]. In 2010, we exceeded 1200 exabytes, in 2015, we were at 7900 exabytes and until 2020, we exceeded 40,000 exabytes [1]. According to another study by IDC in 2018, the data sphere will increase from 33,000 exabytes (amount of data produced until 2018) to 175,000 exabytes in 2025 [2]. The evolution of the data over time is shown in Figure 1.

In the early 2000s, we entered the internet age. Many companies began to understand the value of this data, which was becoming increasingly voluminous. For example, it is possible to predict the onset of a virus such as influenza through human searches on Google. It became clear very quickly that all this data would be used one day. So it was very valuable. There was just not enough time to invent new products using this data for commercial, military or medicinal purposes. Now, with the mass arrival of social networks but also smartphones and connected devices, which were 15 billion in 2015 and which are counted as more than 50 billion today and are predicted to be more than 70 billion in 2025, there will be almost 10 connected devices per human being on average. And all this will generate even more data. All this data is the era of Big Data, which we have only just entered.

The very rapid increase in the amount of data produced brings new challenges in the analysis and storage of this data. Recently, there is a growing interest in key areas such as real-time data mining, which reveal an urgent need to process very large data sets under strict performance constraints. These large masses of data streams are multidimensional and often from different sources. This massive data often brings together content of several types (mixed data), represented by descriptors of different natures: vectors of fixed or variable dimension with real or categorical components, etc. To be able to tap into all the hidden wealth within this avalanche of data, the use of high-quality, high-performance knowledge discovery tools is necessary. Data clustering techniques or algorithms are well known and very powerful knowledge discovery tools for this purpose [3]-[9]. These techniques are shown in Figure 2.

Clustering is an unsupervised learning method for constructing subsets (clusters) whose instances are similar (share the same characteristics) to each other with respect to a given similarity measure and differ from one group to another (dissimilar when they belong to different groups). It is clear that good clustering should achieve high intra-cluster similarity and low inter-cluster similarity [10].

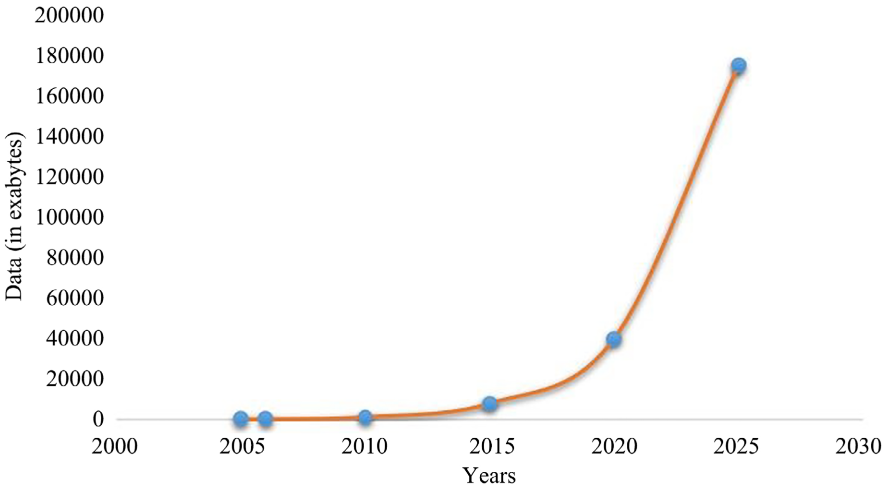

Figure 1. Evolution of data over time.

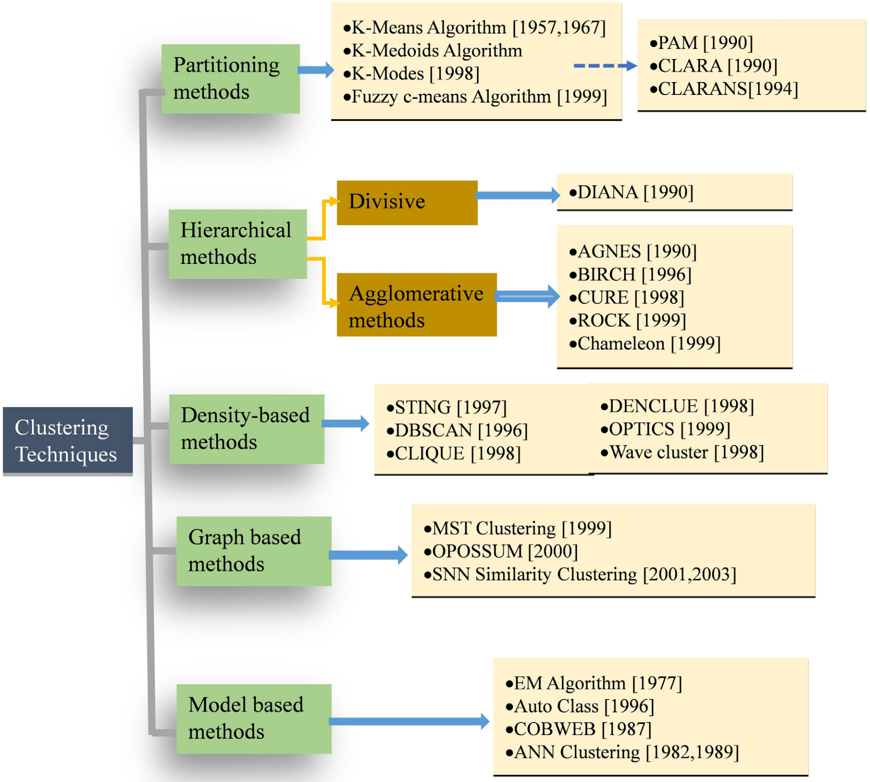

Figure 2. Clustering techniques.

In unsupervised learning, the data is represented as follows:

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{pmatrix}$$

with each row representing an observation. By applying a clustering technique to this data, the output is data grouped by similarity. Clustering will place objects in several groups according to their characteristics. Thus, similar objects will be in the same cluster and dissimilar objects will be in different clusters.

In this paper, we will review some of these methods with higher frequency in previous studies such as K-means algorithm, Fuzzy c-means, BIRCH and Expectation Maximization algorithm used for data clustering. The rest of this paper is organized as follows: Section 3 presents and develops these different algorithms in pseudocode, graphical and/or literal form. The strengths and weaknesses of the presented techniques are also discussed. Section 4 presents the implementation results of these algorithms. Section 5 discusses these results. Section 6 concludes the work.

2. Data Clustering Algorithms in Unsupervised Learning

Clustering is the process of organizing objects (data) into groups based on similar characteristics within the members (data points of the group) [11]. Clustering or clustering algorithms analyze and study the similarities in the data provided by the user in order to classify them into groups. There are several clustering algorithms. For each one, one has to choose the method or technique to be used to measure the similarity between two objects that can be compared to two real points in d-dimensional space. The principle of clustering is to let the machine classify our data according to their similarity. Here are some of these algorithms:

2.1. K-Means Algorithm

The K-means clustering algorithm is one of the best known, most proven, most popular and simplest unsupervised machine learning algorithms [12] [13], which is most often applied to solve clustering problems. The k-means algorithm is an iterative (learning) method to discover a number k of clusters in the input space [14]. The number k is defined a priori [13] [15] [16] and each cluster is represented by a cluster centroid in the feature space. For example, let's imagine that we want to group the data shown in **Figure 3** below into four clusters.

To do this, we will first place four points called centroids (red points in **Figure 4**) at random among our data. Then, we assign each point of the dataset to the nearest centroid which gives us four clusters, then we move each centroid to the middle of its cluster. Then, we start again, we assign each point of our dataset to the nearest centroid and then we move each centroid to the centre of its cluster and we will continue like this until the centroids converge towards an equilibrium position as in the following figure:

Depending on the initial position of the centroids, it is possible that they converge to the wrong positions. The solution is then to run the K-means clustering algorithm several times in a row, changing the initial position of the centroids each time. For each result, the distance between the points of a cluster and the centre of the latter is measured and the solution for which the sum of these distances is the smallest is retained. The K-means algorithm seeks to minimize a

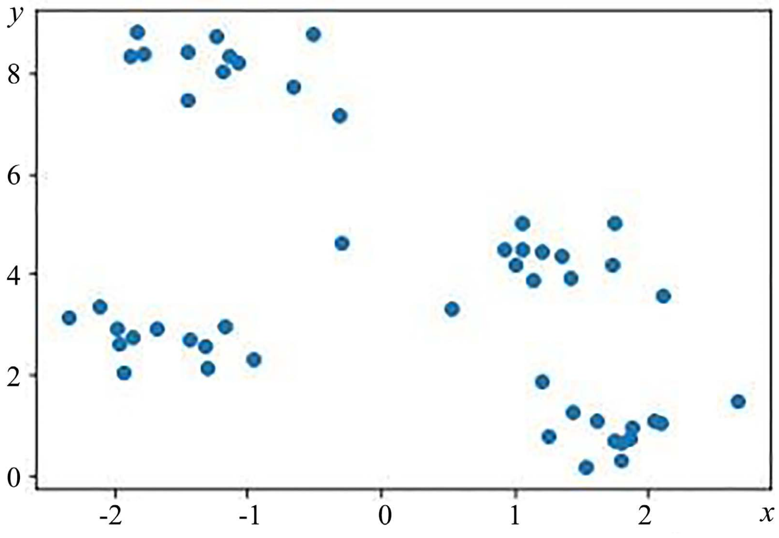

Figure 3. Example of the dataset.

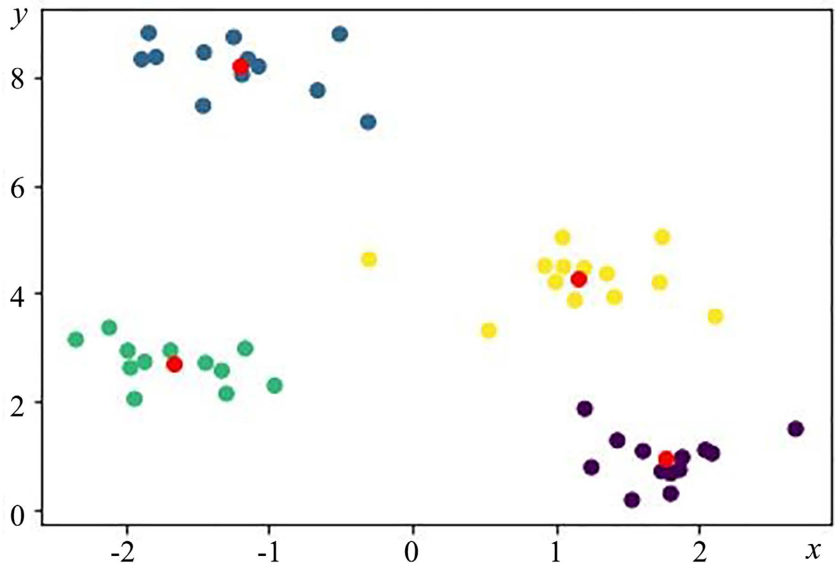

Figure 4. Examples of clustering with K-means.

cost function called inertia, which represents the distance between the points of a cluster and its centre:

$$\sum_{i=0}^{n}\min\left(\left\|x_i-\mu_j\right\|^2\right),\ \mu_j\in C$$

In reality, it seeks to minimize the variance of the clusters. Here is how a K-means clustering algorithm works.

The steps of the K-means algorithm

Step 1. Randomly select k elements (centroids) from the dataset $X=\{x_1,x_2,\cdots,x_n\}$ as a medium $\mu=\mu_1,\mu_2,\cdots,\mu_k$ of the initial clusters;

Step 2. Calculate the Euclidean distance between each element of the dataset and each centroid by: $d_{ij}=\left\|x_i-\mu_j\right\|_2$;

Step 3. Assign each element of the dataset to the nearest centroid. We obtain k clusters;

Step 4. Recalculate the centroids of the k clusters by: $\mu_j=\frac{1}{\left|C_j\right|}\sum_{x_i\in C_j}x_i$; C_j is the j^{th} cluster;

Step 5. Calculate the total mean square error between all the elements of the dataset and their corresponding cluster centroids by:

$$\text{MSE}=\frac{1}{n}\sum_{j=1}^{k}\sum_{x_i\in C_j}\left\|x_i-\mu_j\right\|_2^2;$$

Step 6. If the centroids converge to an equilibrium position, then the results are obtained and exit otherwise return to step 2.

Advantages and disadvantages of the K-means algorithm

K-means is applicable to large data and also any type of data (even textual), by choosing a good notion of distance. The specification of class number, distance/average and initial draw of class centers built on non-existent objects are some of drawbacks of K-means.

2.2. Fuzzy c-Means (FCM) Algorithm

Fuzzy c-means (FCM) is a clustering method that allows a point to belong to two or more clusters, unlike K-means where only one cluster is assigned to each point [17]. This algorithm necessarily assigns a dataset element to one of the clusters by giving them the credibility measure. With FCM, each element of a cluster has the probability of belonging to the other. Therefore, an object does not have an absolute membership on a particular cluster [11]. This method was developed by Dunn in 1973 [18] and improved by Bezdek in 1981 [19] and is based on fuzzy set theory. The Fuzzy c-means procedure [20] is similar to that of K-means. It consists in minimizing the functional *J*:

$$J(B,U,X)=\sum_{i=1}^{C}\sum_{j=1}^{N}u_{ij}^{m}d^{2}(x_j,b_i)$$

If N is the total number of points to be processed; *C:* the number of clusters searched for; $m\in[1,+\infty]$ the degree of fuzziness and *B:* the vector of cluster centres, $U=[u_{ij}]$ will be called a membership degree matrix if and only if:

$$(\forall i\in\{1,\cdots,C\})(\forall j\in\{1,\cdots,N\}),\ u_{ij}\in[0,1],\ 0<\sum_{j=1}^{N}u_{ij}<N,\ \sum_{i=1}^{C}u_{ij}=1.$$

We can then determine B and U using the Lagrange technique. Let us define for each vector x_j, $H(x_j)$ by:

$$H(x_j)=\sum_{i=1}^{C}u_{ij}^{m}d^{2}(x_j,b_i)-\alpha\left(\sum_{i=1}^{C}u_{ij}-1\right),\alpha>0$$

If we cancel the partial derivatives with respect to a u_{ij} and α we get:

$$\frac{\partial H(x_j)}{\partial\alpha}=0\quad\text{and}\quad\frac{\partial H(x_j)}{\partial u_{ij}}=m\cdot u_{ij}^{m-1}\cdot d^{2}(x_j,b_i)-\alpha=0$$

so that if $d^{2}(x_j,b_i)\neq0$, we have: $u_{ij}=\left(\frac{\alpha}{m\cdot d^{2}(x_j,b_i)}\right)^{\frac{1}{m-1}}$ with:

$$\sum_{i=1}^{C}u_{ij}=\left(\frac{\alpha}{m}\right)^{\frac{1}{m-1}}\sum_{i=1}^{C}\left(\frac{1}{d^{2}(x_j,b_i)}\right)^{\frac{1}{m-1}}=1$$

Finally, we have: $u_{ij}=\left[\sum_{k=1}^{C}\left(\frac{d^{2}(x_j,b_i)}{d^{2}(x_j,b_k)}\right)^{\frac{2}{m-1}}\right]^{-1}$

For C and X fixed, if we cancel the partial derivatives $H'(B,g)$ with respect to any direction g of B we obtain:

$$b_i=\frac{\sum_{k=1}^{n}u_{ik}^{m}x_k}{\sum_{k=1}^{n}u_{ik}^{m}}$$

The steps involved in this algorithm are shown in the form of a flow chart in Figure 5.

The FCM algorithm has been widely used to segment brain volumes from one or more multimodalities [21] [22]. Several variants exist for the Fuzzy c-means

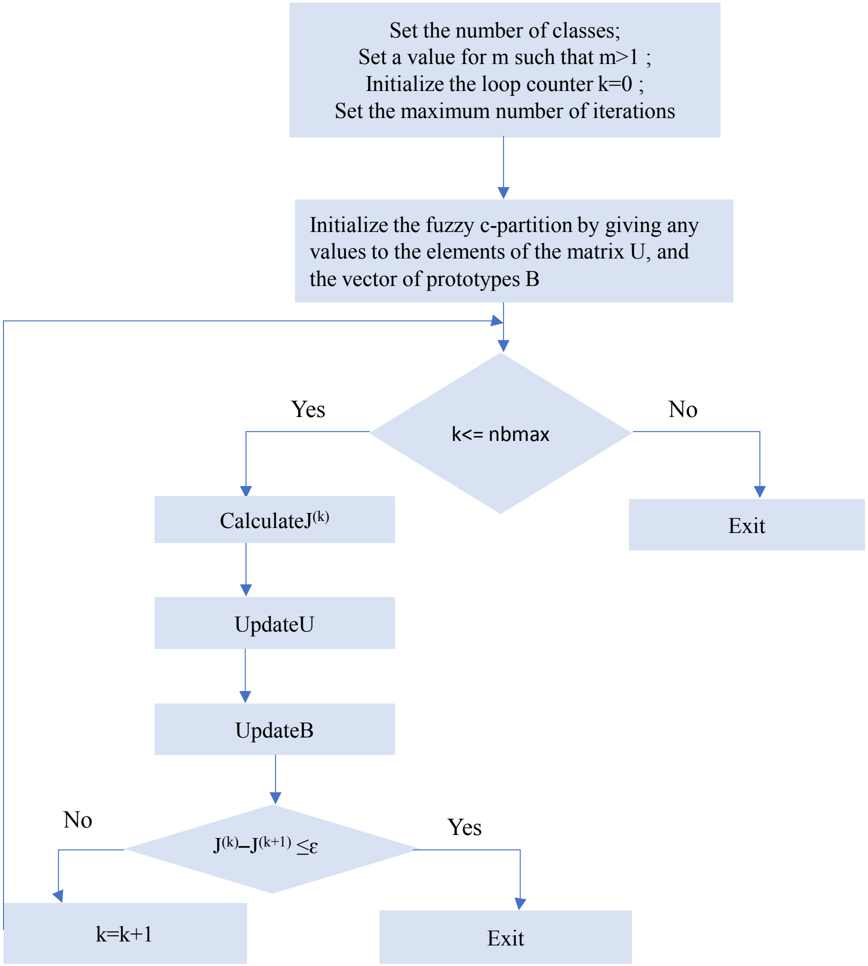
Set the number of classes;
Set a value for m such that m>1 ;
Initialize the loop counter k=0 ;
Set the maximum number of iterations
Initialize the fuzzy c-partition by giving any values to the elements of the matrix U, and the vector of prototypes B
Yes
k<= nbmax
No
Calculate$J^{(k)}$
Exit
UpdateU
UpdateB
No
$J^{(k)}-J^{(k+1)} \le \varepsilon$
Yes
k=k+1
Exit

Figure 5. Fuzzy c-means algorithm.

algorithm. Among them are FcE [23] (fuzzy c-elliptotypes), AFc [24] (adaptive fuzzy c-elliptotypes) [25] [26]. Although this algorithm is widely used in some areas as mentioned, it suffers from several weaknesses. Among them are problems related to the degrees of membership which are relative degrees. In other words, the membership of an individual to a class depends on the membership of this individual to other classes [27]. The membership functions constructed are therefore interdependent. Also, the estimates of the centres of the classes do not correspond to the real or typical centres. Another disadvantage is related to the initialization of the class centres in a random way (step 2) which may influence the convergence of J to a local minimum, so the idea lies in the optimal estimation of these prototypes [28].

2.3. Expectation-Maximization (EM) Algorithm

A general clustering method using statistical principles is to represent the probability density function of the data as a mixture model, which states that the data is a combination of k individual component densities (usually Gaussian), corresponding to k clusters [29]. The EM algorithm is an efficient and popular technique for estimating the parameters of the mixture model [30]. It iteratively refines an initial cluster model to better fit the data and ends with a solution that is locally optimal for the underlying clustering criterion [31]. Like other iterative refinement clustering methods, including the popular k-means algorithm, the EM algorithm is fast and scalable versions are available [32].

In real-world applications of machine learning, it is very common for many relevant features to be available for learning, but for only a small subset of them to be observable. The expectation-maximization algorithm can be used for latent variables (variables that are not directly observable and are in fact inferred from the values of other observed variables). It is in fact the basis of many unsupervised clustering algorithms in the field of machine learning. The EM algorithm is a parametric estimation method within the general framework of maximum likelihood. It is an iterative algorithm that allows to find the maximum likelihood parameters of a probabilistic model when the latter depends on unobservable latent variables. All variables are assumed to be independent of each other and all data are derived from K joint distributions. The Expectation Maximization algorithm is described in detail below:

Inputs: Data $x_i, 1 \le i \le n$, k (number of clusters);

Outputs: Set of parameters $\theta_j, C_j, 1 \le j \le k$ with C_j: the cluster;

Step 1: Randomly select the initial parameters:

$$\theta_j, \omega_j, \left(\mu_j, \sigma_j\right)_{1 \le j \le k};$$

Step 2: Expectation Stage

For each data $x_i, 1 \le i \le n$, calculate:

$$Prob\left(J^{th}\ distribution / x_i, \theta_j\right) = \frac{\omega_j * Prob\left(x_i / \theta_j\right)}{\sum_{j=1}^{k} \omega_j * prob\left(x_i / \theta_j\right)}$$

with:

$$Prob\left(x_i / \theta_j\right) = \frac{1}{\sqrt{2\pi}\sigma_j} \mathrm{e}^{-\frac{\left(x_i - \mu_j\right)^2}{2\sigma_j^2}}$$

Step 3: Maximization stage

Using the $\left(J^{th}\ distribution / x_i, \theta_j\right)$, $1 \le i \le n$, find the new parameter estimate that maximizes the expected likelihood:

$$\mu_j = \sum_{i=1}^{n} x_i \frac{Prob\left(J^{th}\ distribution / x_i, \theta_j\right)}{\sum_{i=1}^{n} Prob\left(J^{th}\ distribution / x_i, \theta_j\right)}$$

Step 4: Repeat steps 2 and 3 if the parameters (centroids) do not change.

Advantages and disadvantages of the Expectation-Maximization algorithm

The advantage of EM over k-means is that it provides a statistical model of the data and is able to handle the associated uncertainties. However, a problem with its iterative nature is the convergence to a local rather than global optimum. It is sensitive to initial conditions and is not robust. While iterative refinement schemes such as k-means and expectation-maximization (EM) are fast and easily adaptable to large databases [32], they can only produce convex groups and are sensitive to parameter initialization.

2.4. BIRCH

BIRCH is a scalable clustering method and is designed for clustering very large datasets by integrating hierarchical clustering and other clustering methods such as iterative partitioning [33]. It overcomes the two difficulties of agglomerative clustering methods:

1) Scalability;

2) The inability to undo what was done in the previous step.

Given N d-dimensional data points in a cluster: $\{\vec{X}_i\}$ where $i = 1, 2, \cdots, N$, we can define the centroid $\vec{X}_0$, the radius R and the diameter D of the cluster as follows:

$$\vec{X}_0 = \frac{\sum_{i=1}^{N} \vec{X}_i}{N};\ R = \sqrt{\frac{\sum_{i=1}^{N} \left(\vec{X}_i - \vec{X}_0\right)^2}{N}};\ D = \sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \left(\vec{X}_i - \vec{X}_j\right)^2}{N(N-1)}}$$

where R is the average distance between member points and the centroid and D the average distance between pairs within a cluster. They reflect the closeness of the cluster around the centroid. We can also define the Euclidean distance from the centroid D_0 and the Manhattan distance of the centroid D_1 of two clusters as follows:

$$D_0 = \sqrt{\left(\vec{X}_{0_1} - \vec{X}_{0_2}\right)^2}$$

$$D_1 = \left|\vec{X}_{0_1} - \vec{X}_{0_2}\right| = \sum_{i=1}^{d} \left|\vec{X}_{0_1}^{(i)} - \vec{X}_{0_2}^{(i)}\right|$$

when their centroids $\vec{X}_{0_1}$ and $\vec{X}_{0_2}$ are given.

The average inter-cluster distance D_2 the average intra-cluster distance D_3 and the distance of increase in variance D_4 when we have N_1 d-dimensional data points in the cluster $\{\vec{X}_i\}$ with $i = 1, 2, \cdots, N_1$, and N_2 data points in another cluster $\{\vec{X}_j\}$ with $j = N_1 + 1, N_1 + 2, \cdots, N_1 + N_2$ are defined as follows:

$$D_2 = \sqrt{\frac{\sum_{i=1}^{N_1} \sum_{j=N_1+1}^{N_1+N_2} \left(\vec{X}_i - \vec{X}_j\right)^2}{N_1 N_2}}$$

$$D_3 = \sqrt{\frac{\sum_{i=1}^{N_1+N_2} \sum_{j=1}^{N_1+N_2} \left(\vec{X}_i - \vec{X}_j\right)^2}{\left(N_1 + N_2\right)\left(N_1 + N_2 - 1\right)}}$$

$$D_4 = \sum_{k=1}^{N_1+N_2} \left(\vec{X}_k - \frac{\sum_{l=1}^{N_1+N_2} \vec{X}_l}{N_1 + N_2}\right)^2 - \sum_{i=1}^{N_1} \left(\vec{X}_i - \frac{\sum_{l=1}^{N_1} \vec{X}_l}{N_1}\right)^2 - \sum_{j=N_1+1}^{N_1+N_2} \left(\vec{X}_j - \frac{\sum_{l=N_1+1}^{N_1+N_2} \vec{X}_l}{N_2}\right)^2$$

To summarize the cluster representations, BIRCH uses two methods, namely the clustering feature (CF) and the clustering feature tree (CF tree). According to

[34], when we have N d-dimensional data points in the cluster, $\{\vec{X}_i\}$, the clustering feature (CF) vector of the cluster is defined as $\text{CF}=(N,\vec{LS},SS)$ with:

- N: the number of data points in the cluster;
- $\vec{LS}$: the linear sum of the N data points (*i.e.*, $\sum_{i=1}^{N}\vec{X}_i$);
- SS: The square sum of the N data points (*i.e.*, $\sum_{i=1}^{N}\vec{X}_i^2$).

A CF tree is a height-balanced tree that stores the clustering characteristics for hierarchical clustering and has two parameters namely the branching factor B and the threshold T. The advantage of the BIRCH algorithm is that it can find a good clustering with a single scan and improve the quality with a few more scans, but unfortunately it only processes numerical data. The execution steps of the BIRCH algorithm are shown in Figure 6.

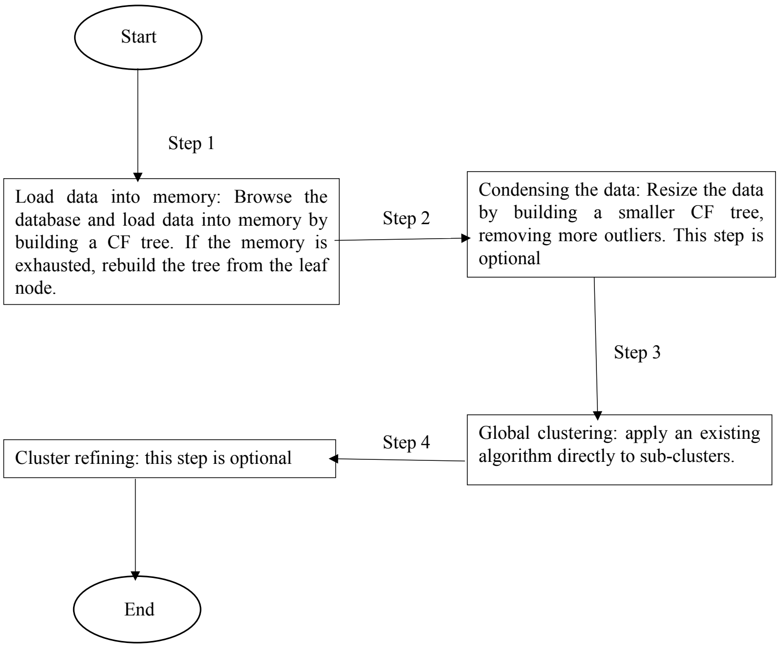
Start
Step 1
Load data into memory: Browse the database and load data into memory by building a CF tree. If the memory is exhausted, rebuild the tree from the leaf node.
Step 2
Condensing the data: Resize the data by building a smaller CF tree, removing more outliers. This step is optional
Step 3
Global clustering: apply an existing algorithm directly to sub-clusters.
Step 4
Cluster refining: this step is optional
End

Figure 6. Steps in the BIRCH algorithm.

3. Implementation Results of These Algorithms

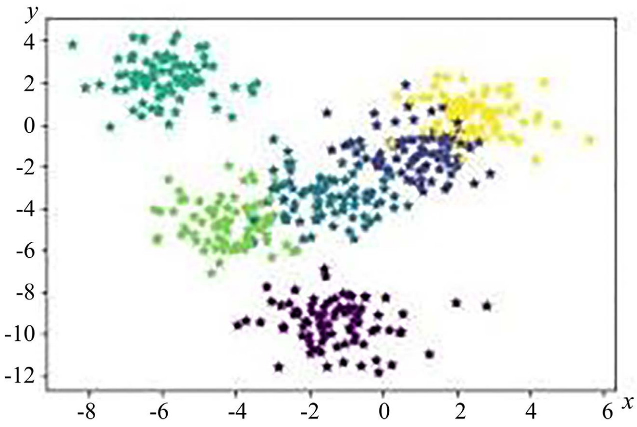

Figure 7. Data set.

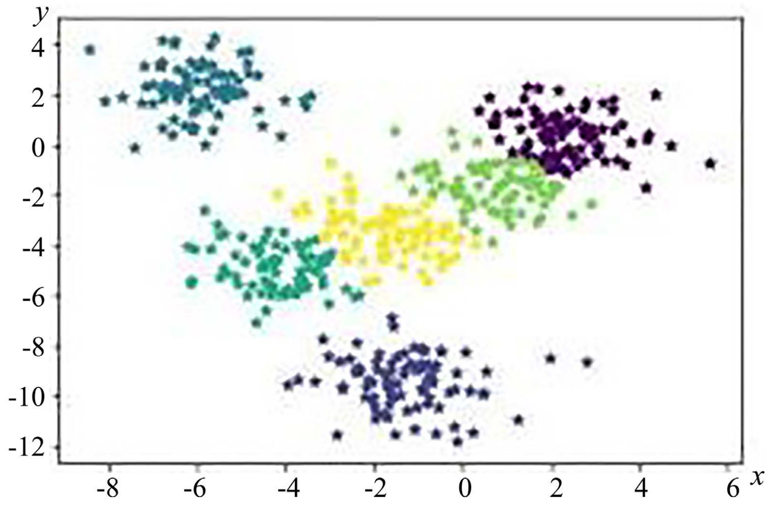

Figure 8. K-means clustering algorithm.

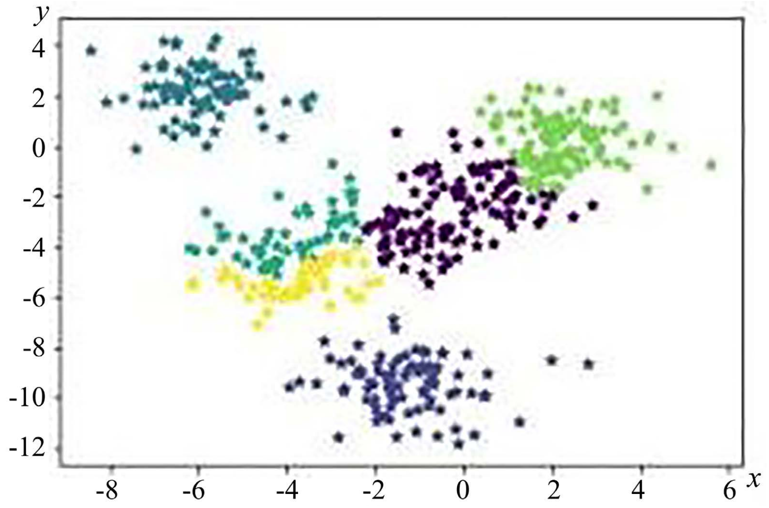

Figure 9. BIRCH clustering algorithm.

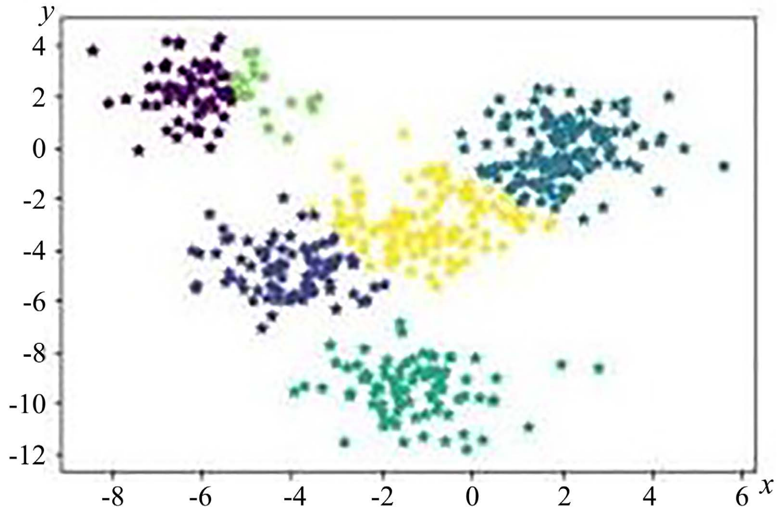

Figure 10. EM clustering algorithm.

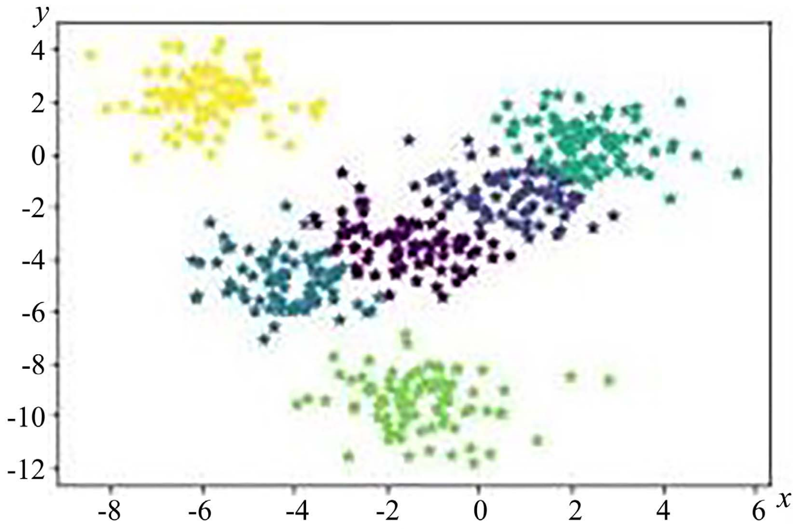

Figure 11. FCM clustering algorithm.

4. Discussion

We have just applied the four algorithms on the same data set (**Figure 7**) to see the differences in the results obtained. By analyzing these results, we can see that for some algorithms, there is a small difference in the classification while for others, similarities are clearly evident. For example, as presented in **Figure 8**, with the K-Means clustering algorithm, we can see that the clusters are clearly separated in such a way that we can even think that there is an imaginary line separating each pair of clusters, which is not the case for the BIRCH clustering algorithm as shown in **Figure 9**. With the FCM clustering algorithm in **Figure 11**, the results are almost similar to the results obtained using the K-Means algorithm. The clusters are well compacted and separated. The results of the EM algorithm in **Figure 10** are also close to the results of the K-Means, EM and BIRCH algorithm. In general, taking into account the results obtained using these different algorithms on the same data, it can be seen that there is not a big difference. Therefore, it cannot be said that one algorithm is better than another with respect to the results obtained here. Meanwhile, as highlighted while describing each clustering method, each one has its advantages and drawbacks. Consequently, the choice of a specific method depends on the kind of data and the aimed outcomes and/or results.

5. Conclusion

Clustering of large and numerous data is a data classification method used to extract valuable knowledge that can guide decision makers and business managers. The topicality of this work lies in the deep review of the evolution of the volumes, nature and form of data, which, with their structural and functional complexity, render the classical methods of data management of transactional systems almost obsolete. The new methods, namely those based on unsupervised learning, significantly solve a range of problems encountered when using classical methods. In particular, they allow for the improved capitalization of data and knowledge in this digital era focused on the knowledge economy. The study of modern clustering methods has allowed us to identify a real mapping of these methods from the perspective of their design. The implementation of K-means, Fuzzy c-means, BIRCH and Expectation-Maximization methods shows that there is practically no one method that is more effective and efficient than another. Thus, a rational and therefore sound approach may require the combination of a number of these methods in order to take advantage of the benefits that one or the other method offers. Future research will concentrate on studying, stimulating and analyzing how the combination of two or more clustering methods improves and reinforces the model's efficacy, efficiency and rationality.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] John, G. and David, R. (2012) The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. IDC IVIEW, Sponsored by EMC Corporation. https://www.cs.princeton.edu/courses/archive/spring13/cos598C/idc-the-digital-universe-in-2020.pdf

[2] David, R., John, G. and John, R. (2018) The Digitization of the World, from Edge to Core. An IDC White Paper-#US44413318, Sponsored by Seagate. https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf

[3] Williams, P., Soares, C. and Gilbert, J.E. (2012) A Clustering Rule Based Approach for Classification Problems. *International Journal of Data Warehousing and Mining*, **8**, 1-23. https://doi.org/10.4018/jdwm.2012010101

[4] Priya, R.V. and Vadivel, A. (2012) User Behaviour Pattern Mining from Weblog. *International Journal of Data Warehousing and Mining*, **8**, 1-22. https://doi.org/10.4018/jdwm.2012040101

[5] Kwok, T., Smith, K.A., Lozano, S. and Taniar, D. (2002) Parallel Fuzzy c-Means Clustering for Large Data Sets. In: Monien, B. and Feldmann, R. Eds., *Euro-Par* 2002: *Euro-Par* 2002 *Parallel Processing*, Springer, Berlin, 365-374. https://doi.org/10.1007/3-540-45706-2_48

[6] Kalia, H., Dehuri, S. and Ghosh, A. (2013) A Survey on Fuzzy Association Rule Mining. *International Journal of Data Warehousing and Mining*, **9**, 1-27. https://doi.org/10.4018/jdwm.2013010101

[7] Daly, O. and Taniar, D. (2004) Exception Rules Mining Based on Negative Association Rules. In: Laganá, A., Gavrilova, M.L., Kumar, V., Mun, Y., Tan, C.J.K. and Gervasi, O. Eds., *Computational Science and Its Applications—ICCSA* 2004, Springer, Berlin, 543-552. https://doi.org/10.1007/978-3-540-24768-5_58

[8] Ashrafi, M.Z., Taniar, D. and Smith, K.A. (2007) Redundant Association Rules Reduction Techniques. *International Journal of Business Intelligence and Data Mining*, **2**, 29-63. https://doi.org/10.1504/IJBIDM.2007.012945

[9] Taniar, D., Rahayu, W., Lee, V.C.S. and Daly, O. (2008) Exception Rules in Association Rule Mining. *Applied Mathematics and Computation*, **205**, 735-750. https://doi.org/10.1016/j.amc.2008.05.020

[10] Havens, T.C., Bezdek, J.C. and Palaniswami, M. (2013) Scalable Single Linkage Hierarchical Clustering for Big Data. 2013 *IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Melbourne, 2-5 April 2013, 396-401. https://doi.org/10.1109/ISSNIP.2013.6529823

[11] Abhishek, S. (2018) Most Popular Clustering Algorithms Used in Machine Learning. https://analyticsindiamag.com/most-popular-clustering-algorithms-used-in-machine-learning/

[12] Lam, D. and Wunsch, D.C. (2014) Clustering. *Academic Press Library in Signal Processing*, **1**, 1115-1149. https://doi.org/10.1016/B978-0-12-396502-8.00020-6

[13] MacQueen, J. (1967) Some Methods for Classification and Analysis of Multivariate Observations. In: Le Cam, L.M., Neyman, J., and Scott, E.L., Eds., Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Oakland, 281-297.

[14] Artúr, I.K., Róbert, F. and Galambos, P. (2018) Unsupervised Clustering for Deep Learning: A Tutorial Survey. *Acta Polytechnica Hungarica*, **15**, 29-53. https://doi.org/10.12700/APH.15.8.2018.8.2

[15] Wu, X.D., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., *et al.* (2008) Top 10 Algorithms in Data Mining. *Knowledge and Information Systems*, **14**, 1-37. https://doi.org/10.1007/s10115-007-0114-2

[16] Tapas, K., David, M.M., Nathan, S.N., Christine, D.P., Ruth, S. and Angela, Y.W. (2002) An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 881-892. https://doi.org/10.1109/TPAMI.2002.1017616

[17] Amit, S., *et al.* (2017) A Review of Clustering Techniques and Developments. *Neurocomputing*, **267**, 664-681. https://doi.org/10.1016/j.neucom.2017.06.053

[18] Dunn, J.C. (1973) A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, **3**, 32-57. https://doi.org/10.1080/01969727308546046

[19] Bezdek, J.C. (1981) Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York. https://doi.org/10.1007/978-1-4757-0450-1

[20] Xu, R. and Wunsch, D. (2005) Survey of Clustering Algorithms. *IEEE Transaction on Neural Networks*, **16**, 645-678. https://doi.org/10.1109/TNN.2005.845141

[21] Chen, W. and Giger, M. (2006) A Fuzzy C-Means (FCM)-Based Approach for Computerized Segmentation of Breast Lesions in Dynamic Contrast-Enhanced MR Images. *Academic Radiology*, **13**, 63-72. https://doi.org/10.1016/j.acra.2005.08.035

[22] Jiang, L. and Yang, W.H. (2003) A Modified Fuzzy c-Means Algorithm for Segmentation of Magnetic Resonance Images. *Proceedings of the 7th International Conference on Digital Image Computing*: *Techniques and Applications*, DICTA 2003, Sydney, 10-12 December 2003, 225–232.

[23] Bezdek, J., Keller, J., Pal, N. and Krisnapuram, R. (1995) Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Kluwer Academic Publishers, New York.

[24] Dave, R.N. (1992) Boundary Detection through Fuzzy Clustering. *IEEE Interna-*

tional Conference on Fuzzy Systems, San Diego, 8-12 March 1992, 127-134.

[25] Kang, J.Y., Min, L,Q., Luan, Q.X., Li, X. and Liu, J.Z. (2009) Novel Modified Fuzzy c-Means Algorithm with Applications. *Digital Signal Processing*, **19**, 309-319.

[26] Berget, I., Mevik, B.H. and Næs, T. (2008) New Modifications and Applications of Fuzzy c-Means Methodology. *Computational Statistics & Data Analysis*, **52**, 2403-2418. https://doi.org/10.1016/j.csda.2007.10.020

[27] Barra. V. (1999) Segmentation floue des tissus cérébraux en IRM 3D: Une approche possibiliste versus autres méthodes. Master's Thèse, Universite Blaise Pascal, Clermont-Ferrand.

[28] Moussa, S., Lyazid, T. and Abdelouaheb, M. (2008) Nouvelle variante de l'algorithme FCM Appliquée à la Segmentation D'images IRM Cérébrales. *MCSEAI*, Oran, 28-30 April 2008, 4 p.

[29] Pabitra, M., Sankar, K.P. and Siddiqi, M.A. (2003) Non-Convex Clustering Using Expectation Maximization Algorithm with Rough Set Initialization. *Pattern Recognition Letters*, **24**, 863-873. https://doi.org/10.1016/S0167-8655(02)00198-8

[30] Cherkassky, V. and Mulier, F. (1998) Learning from Data: Concepts, Theories and Methods. John Wiley, New York.

[31] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977) Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*: *Series B*, **39**, 1-22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

[32] Bradley, P., Fayyad, U. and Reina, C. (1999) Scaling EM (Expectation Maximization) Algorithm to Large Databases. Microsoft Research Technical Report, MSR-TR-98-35. https://www.researchgate.net/publication/2240573_Scaling_EM_Expectation-Maximization_Clustering_to_Large_Databases

[33] Study Materials: APJ Abdul Kalam Technological University. https://www.marian.ac.in/public/images/uploads/DMWH%20M6.pdf

[34] Zhang, T., Ramakrishnan, R. and Livny, M. (1996) BIRCH: An Efficient Data Clustering Method for Very Large Databases. *ACM SIGMOD Record*, **25**, 103-114. https://doi.org/10.1145/235968.233324