Fuzzy Partition Technique for Clustering Big Urban Dataset

Ahmad AlShami
Computing and Information Science
Higher Colleges of Technology
Fujairah, United Arab Emirates
Email: Ahmad.AlShami@hct.ac.ae

Weisi Guo
School of Engineering
The University of Warwick
Coventry, United Kingdom
Email: Weisi.Guo@warwick.ac.uk

Ganna Pogrebna
Warwick Institute for the Science of Cities
The University of Warwick
Coventry, United Kingdom
Email: Ganna.Pogrebna@warwick.ac.uk

***Abstract*—Smart cities are collecting and producing massive amount of data from various data sources such as local weather stations, LIDAR data, mobile phones sensors, Internet of Things (IoT) etc. To use such large volume of data for potential benefits, it is important to store and analyse data using efficient and effective big data algorithms. However, this can be problematic due to many challenges. This article explores some of these challenges and tested the performance of two partition algorithms for clustering such Big Urban Datasets. Two handy clustering algorithms the K-Means vs. the Fuzzy c-Mean (FCM) were put to the test. The purpose of clustering urban data is to categorize it into homogeneous groups according to specific attributes. Clustering Big Urban Data in compact format represents the information of the whole data and this can benefit researchers to deal with this reorganised data much efficiently. To achieve this end, the two techniques were utilised against a large set of Lidar data to show how they perform on the same hardware set-up. Our experiments conclude that FCM outperformed the K-Means when presented with such type of dataset, however the latter is less demanding on the hardware utilisation.**

***Keywords*—Big Data; Fuzzy c-Mean; Hardware Utilisation; K-Means; LIDAR; Smart City**

I. Introduction

Many ongoing and recent researches and development in computation and data storing technologies have contributed to production of the Big Data phenomena. The challenges of Big Data are due to the 5V's which are: Volume, Velocity, Variety, Veracity and Value to be gained from the analysis of Big Data [1]. From the survey of the literature, there is an agreement between data scientists about the general attributes that characterise Big Data 5V's which can be summed as follows:

- Very large data mainly in Terabytes/Petabytes/Exabyte's of data (Volume).
- Data can be found in structured, unstructured and semi-structured forms (Variety).
- Often incomplete data and inaccessible.
- Data sets extraction should be from reliable and verified sources.
- Data can be streaming at very high speed (Velocity).
- Data can be very complex with interrelationships and high dimensionality.
- Data may contain few complex interrelationships between different elements.

The challenges of Big Data in general are an ongoing thing and the problems is growing every year. A report by Cisco [2], estimated that by the end of 2017, annual global data traffic will reach 7.7 Zettabytes. The global internet traffic will be three times over the next five years. Overall, the global data traffic will grow at a Compound Annual Growth Rate (CAGR) of 25% by the year 2017. It is essential to take steps toward tackling these challenges because it can be predicted that a day will come when Big Data tools will become obsolete in front of such enormous data flow.

This Paper is organized as follows: Section II includes related work on the characteristics of Big Data and the work conducted for the purpose of Big Data analytic. Section III, gives general overview of the available clustering techniques and a pseudo description for the K-Means and FCM are given. The utilised experiments, comparative analysis and our findings are highlighted in Section IV. Conclusion and future directions are summed up in Section V.

II. Related Work

Big Data analytic is helping governments to control security threats, and to have better management and services. According to [3] the US National Security Agency is able to analyse the calls and text message data of hundreds of millions of mobile users. For this purpose graphic analysis is ideal to identify the connections which requires powerful I/O capacity. In some cases, Big Data visualizations and infographic can also be helpful. According to [4] Big Data can help fraud detection as it looks for patterns and strange activities. Using Big Data analysis and visualizations, security agents can find a pattern for unusual threats in real time, also this enables them to find the location with predicted escape routes, and they can act much faster in response to the threats. By using historical data, future cases of fraud can be prevented, hence in May 2012, the US Medicare Fraud Strike Force recovered 452 million from fraud billings and it was found that at least 8 percent of the annual health care expenditures are attributed to fraud.

When it comes to the tools and techniques used to handle Big Data, [6] developed a new technique called "PROXIMUS" which is efficient in clustering and finding patterns in a very large dataset. Another approach by [9] show that decision tree

can be used on a large dataset to extract rules generated from independent and large number of subset of data. The authors in [10] combined genetic algorithm and decision tree to improve the performance and efficiency of the computation. Different clustering techniques to analyse the different sizes of data sets using GLC++ as a new algorithm which was developed by [11] to deal with large different types of datasets. In addition, More techniques like structural coding, frequencies, co-occurrence and graph theory, data reduction techniques, hierarchical clustering techniques, multidimensional scaling were defined in data reduction techniques such as Principle Component Analysis (PCA) for large qualitative dataset to analyse the pattern as required [13]. Two soft computing techniques the Self-Organizing Map (SOM) and learning vector quantization (LVQ) were compared by [15] to categorize large dataset into smaller sets to improve the computation time.

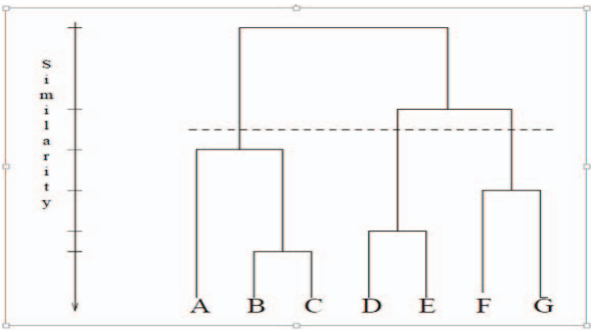
III. Clustering Methods

Researchers are busy dealing with many types large data sets, the concern here is to wither introduce new algorithms or use the existing algorithms to suit large datasets by focusing on the data itself to suit the available algorithms. Currently, two approaches are predominant: First, is known as "Scaling-Up" which focuses the efforts on the enhancement of the available algorithms. This approach risks them becoming useless for tomorrow, as the data continues to grow. Hence, to deal with continuously growing in size datasets, it will be necessary to frequently scale up algorithms as the time moves on. The second approach is to "Scale-Down" or to skim the data itself, and to use existing algorithms on the skimmed version of the data after reducing its size. The scaling down of data may also risk the loss of valuable information due the summarising and size reductions techniques. But, still it is argued that using the "scaling down" technique may only risk the information that is comparatively unimportant or redundant. Since there is still a great scope for the research in both areas, this article focuses on the scale-down of data sets by comparing clustering techniques.

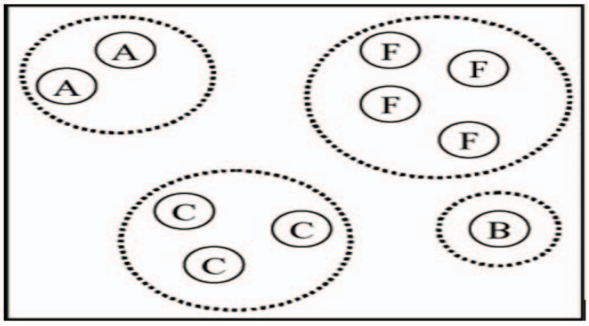
Clustering is defined as the process of grouping a set of items or objects which have same attributes or characteristics in the same group called a cluster which may differ from another group [12]. Clustering can be very useful for between cluster separation, within cluster homogeneity and for good representation of data by its centroid. These can be applied to different fields such as Biology to find groups of genes which have same functions or similarities. It is also used in Medicine to find patterns in symptoms of disease and in Business to find and target potential customers. The authors in [16], [20] and [21] agree on two major methods that are more suitable for Big Data as they require less time and space to converge. These methods can be summarised as follows:

1) **Hierarchical clustering:** A technique which tends to create leaf-nodes (hierarchy) type of clusters according to the inter-mediate nodes as a medium of proximity. There are two types of hierarchical clustering:
   - Agglomerative: It is a move up the hierarchy by placing each pattern in its own cluster, then pairs of clusters get combined.
   - Divisive: Here the move is down the hierarchy where the patterns starts from a super-cluster, and then splits it to smaller clusters as it moves down the hierarchy.

   The major hierarchical techniques are BIRCH, CHAMELEON, CURE, ROCH etc. However, these methods does not scale well for Big Data as it suffers from the following disadvantages:
   - Time complexity: It is time consuming and the time required can be estimated using $O(n^2 * logn)$, where n is the total number of records.
   - Space Complexity: Most hierarchical techniques require storing a similarity matrix of size $O(n^2)$.
   - Irreversible syndrome: All actions of a step-up or a split down cannot be reversed, therefore original dataset cannot be back-tracked [20].
2) **Partitional clustering:** A simple grouping for a set of data items or objects into sub-groups (clusters) by moving the data objects from one cluster to another starting from an initial groups. K-means and other clustering algorithms has been heavily investigated in the literature [23] and [24]. Other partitional algorithms such as CLARA and PAM were also investigated by [7], [21].



(a) Hierarchical



(b) Partitional

Fig. 1: Illustrations of Hierarchical and Partitional clustering. Hierarchical (a), Partitional (b)

Fig. 1 -a and -b separately illustrate the above mentioned types of clusters.

A. Compared Techniques: K-Means vs. Fuzzy c-Means

To highlight the advantages to everyday computing for Big Data and to avoid the above mentioned disadvantages for the hierarchical clustering techniques, this article is focusing on comparing two trendy and computationally attractive partitional techniques which are explained below:

1) K-Means Clustering: This is a widely used clustering algorithm. It partition a data set into K clusters $(C_1, C_2, ..., C_K)$, represented by their arithmetic means called the "centroid" which is calculated as the mean of all data points (records) belonging to certain cluster:

$$\mu_k = \frac{1}{n_k} \sum_{q=1}^{n_k} x_q \tag{1}$$

where n_k is the number of records belonging to cluster k and μ_k is the arithmetic mean of the cluster k.

In each iteration, each data point is assigned to its nearest cluster centroid by using a distance measure such as Euclidean or Manhattan.

2) Fuzzy c-Means clustering: FCM was introduced by [18] and it is derived from the explained K-means concept for the purpose of clustering datasets, but it differs in that the object may belong to more than one cluster with degrees of belonging. However, it is possible that an object may belong to more than one cluster according to its degree of membership, which is also calculated on the bases of distances (usually the Euclidean) between the data points and cluster centre. The FCM clustering is obtained by minimizing the objective function at each iteration, an objective function is minimized to find the best location for the clusters and its values are returned in objective function. For a data set represented as $X = \{x_1, x_2, \ldots, x_j \ldots, x_n\} \subset R^s$ into c clusters, where $1 < c < n$; the fuzzy clusters can be characterized by a $c \times n$ membership function matrix U, whose entries satisfy the following conditions:

$$\sum_{i=1}^{c} u_{i,j} = 1, \qquad j = 1, 2, \ldots, n \tag{2}$$

$$0 < \sum_{j=1}^{n} u_{i,j} < n, \qquad i = 1, 2, \ldots, c \tag{3}$$

where $u_{i,j}$ is the grade of membership for x_j data entry in the ith cluster. Cluster centres are determined initially at the learning stage. Then, the classification is made by comparison of distance between the data points and cluster centres. Clusters are obtained by the minimisation of the following cost function via an iterative scheme.

$$J(U,V) = \sum_{j=1}^{n} \sum_{i=1}^{c} (u_{i,j})^2 \left\| x_j - v_i \right\| \tag{4}$$

where $V = \{v_1, v_2, \ldots, v_i, \ldots v_c\}$ are c vectors of cluster centres with v_i representing the centre for ith cluster.

To calculate the centre of each cluster, the following iterative algorithm is used.

1) Estimate the class membership U.
2) Calculate vectors of cluster centres $V = \{v_1, v_2, \ldots, v_i, \ldots v_c\}$ using the following expression:

$$v_i = \frac{\sum_{j=1}^{n} (u_{i,j})^2 x_j}{\sum_{j=1}^{n} (u_{i,j})^2} \qquad i = 1, 2, \ldots, c \tag{5}$$

3) Update the class membership matrix U with:

$$u_{i,j} = \frac{1}{\sum_{r=1}^{c} \left(\frac{\left\| x_j - v_i \right\|}{\left\| x_j - v_r \right\|} \right)^2} \; i = 1, \ldots, c; \;\; j = 1, \ldots, n \tag{6}$$

4) If control error defined as the difference between two consecutive iterations of the membership matrix U is less than a pre-specific value, then the process can stop. Otherwise process will repeat again from step 2.

After a number of iterations, cluster centres will satisfy the minimisation of the cost function J to a local minimum [19].

IV. Experiments and Analysis

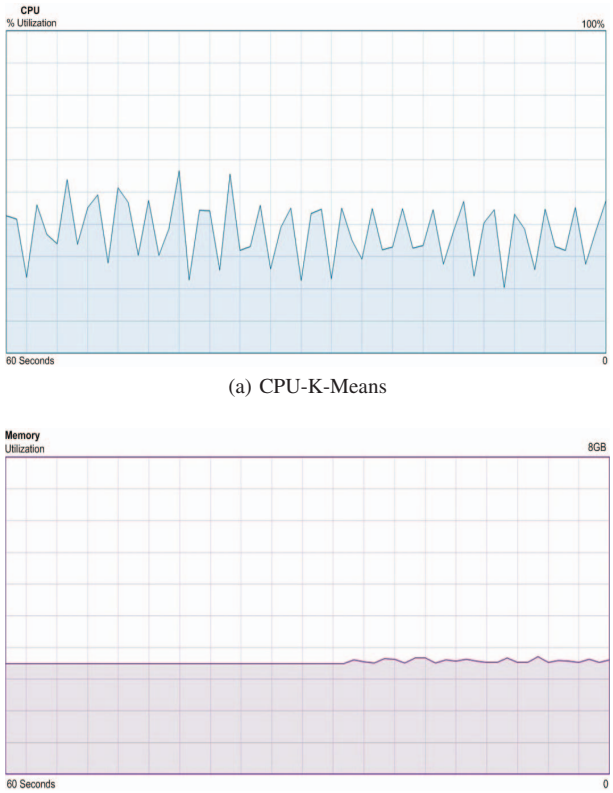
A. Experiments Set-up

The experiments are done to compare and illustrate how the candidate K-Means and FCM clustering techniques cope with clustering Big Urban Data set using a handy computer hardware. The experiment were performed using an AMD 8320, 4.1 GHz, 8 core processor with 8 GB of RAM and running a 64-bit Windows 8.1 OS. The algorithms were implemented against a a LIDAR data points [5], taken for our campus location at Latitude: 52.23° - 52.22° and Longitude: 1.335° - 1.324°. This location represents the University of Warwick main campus with an initialization of 1000000 x 1000 digital surface data points. Fig. 2 shows the boundary of the area represented by the selected dataset.

Fig. 2: Boundary of the area represented by the selected LIDAR dataset.

1) K-Means Clustering: K-Means clustering technique is applied to the dataset starting with a small cluster number $K = 5$ and gradually increased to reach $K = 25$ clusters. Table I lists a summary of the statistics of elapsed time and resources

used for K-Means algorithm to converge. Fig. 3 shows how on average the used hardware fared to obtain the desired number of K clusters.



(a) CPU-K-Means

(b) RAM-K-Means

Fig. 3: Average CPU and Memory usage during K-Means execution.(a) CPU, (b) RAM.

2) FCM Clustering: FCM clustering technique was also applied to same generated dataset with cluster number started with 5 and gradually increased to reach 25 clusters. Sample of these runs which was created to give the final clusters after iterations are listed below.

- $Iteration count = 1$, obj. fcn = 2203700.363302
- $Iteration count = 2$, obj. fcn = 1666584.382950
- $Iteration count = 3$, obj. fcn = 1666584.316099
- $Iteration count = 4$, obj. fcn = 1666584.316098

$Elapsed time = 42.190564 seconds$

When the number of cluster is increased to (10) the result is as follows:

(a) CPU-FCM

(b) RAM-FCM

Fig. 4: Average CPU and Memory usage during FCM execution.(a) CPU, (b) RAM.

- $Iteration count = 1$, obj. fcn = 1109243.044572
- $Iteration count = 2$, obj. fcn = 833296.828286
- $Iteration count = 3$, obj. fcn = 833296.797609
- $Iteration count = 4$, obj. fcn = 833296.797608

$Elapsed time = 83.577006 seconds$

Hence, it is noted the direct relation between the number of clusters and the time of operations; while memory and CPU usage remains the same as compared to first test run. Table II lists summary of the main time and resources it took the FCM algorithm to converge for the different number of assigned clusters.

Fig. 4-a and Fig. 4-b show the CPU and RAM usage while executing the large dataset with FCM clustering function.

TABLE I: Time elapsed and resources used for K-Means clustering.

Clusters counts	Time/Seconds	CPU used	RAM used
5	161.178	21% of 4.0 GHz	36% of 8.0 GB
10	244.642	27% of 4.0 GHz	42% of 8.0 GB
15	338.345	36% of 4.0 GHz	47% of 8.0 GB
20	409.618	48% of 4.0 GHz	53% of 8.0 GB
25	484.013	55% of 4.0 GHz	58% of 8.0 GB
Average	**327.558**	**37.4%**	**47.2%**

TABLE II: Time elapsed and resources used for FCM clustering.

Clusters counts	Time/Seconds	CPU used	RAM used
5	42.190	56% of 4.0 GHz	65% of 8.0 GB
10	83.577	59% of 4.0 GHz	67% of 8.0 GB
15	127.848	65% of 4.0 GHz	75% of 8.0 GB
20	168.994	67% of 4.0 GHz	87% of 8.0 GB
25	214.995	69% of 4.0 GHz	91% of 8.0 GB
Average	**127.520**	**63.2%**	**77.0%**

B. Comparative Analysis

By revisiting the numbers in Table I and Table II, it is clear the difference in the needed time between the utilised algorithms. The lowest time measured for FCM to regroup the data into 5 clusters was recorded at 42.18 seconds while it took K-Means 161.17 seconds to form the same number of clusters. The highest time recorded for K-Means to converge was 484.01 seconds, while it took FCM 214.995 seconds to cluster the same dataset. Hence, There is a high positive correlation between the time and the number of clusters assigned, as the number of clusters count increases so does the time complexity for both algorithms as depicted in Fig. 5.

On average FCM used up between $5-7$ out of the eight available cores, with 63.2 percent of the CPU processing power and 77 percent of the RAM memory. The K-Means on the other hand utilised between $4-6$ with the rest remain as idle cores with an average of 37.4 percent of the CPU processing power and 47.2 percent of the RAM memory.

Overall, both algorithms are scalable to deal with Big Data, but, FCM is fast and would make an excellent clustering algorithm for everyday computing. In addition, it would offer some extra added advantages such as its ability to handle different data types [20]. Also, this fuzzy partitioning technique and due to its fuzzy capability, FCM could produce a better quality of the clustering output [21] which could benefit many data analysts.

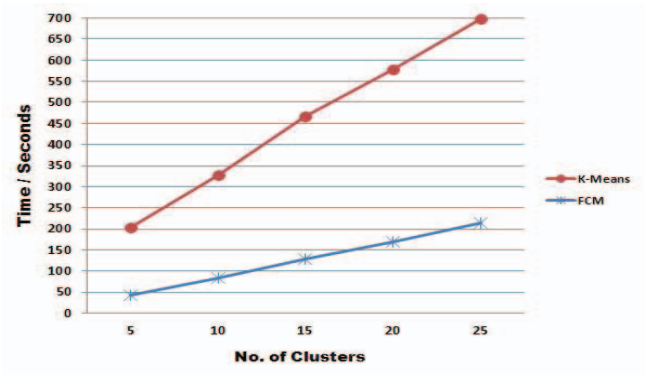


Fig. 5: Comparative plot for elapsed time vs. number of clusters for K-Means and FCM executions.

V. Conclusions and Future Work

A comparative case study for clustering Big Urban Data set using handy and simple techniques is proposed. The K-Means and FCM were tested to cluster a Big Data set hosted on a PC for everyday computing. The presented techniques can be instantly mobilised as a robust methods to handle partitional clustering for a large dataset with ease. However, FCM would be a better choice if speed and quality are priority. In the near future we plan to focus our attention on the quality of the clusters produced here and to compare more clustering techniques against higher dimensionality datasets.

References

[1] Zhai, Y., Ong, Y., & Tsang, I. (2014). The Emerging "Big Dimensionality". Computational Intelligence Magazine, IEEE, 9(3), 14-26.

[2] Cisco Global Cloud Index:Forecast & Methodology, 2012-2017.

[3] Harris D., Here's how the NSA analyzes all that call data. Available: http://gigaom.com/2013/06/06/heres-how-the-nsa-analyzes-all-that-call-data. Last accessed May 3rd 2015.

[4] Cull B., 3 ways big data is transforming government, 8, 2013.

[5] LIDAR Digital Terrain Model. Available upon license, The Environment Agency: http://data.gov.uk/dataset/lidar-digital-surface-model. Last accessed 2nd April 2015.

[6] Mehmet Koyuturk, Ananth Grama, and Naren Ramakrishnan, "Compression, Clustering, and Pattern Discovery in very High-Dimensional Discrete-Attribute Data Sets", IEEE Transactions On Knowledge And Data Engineering, April 2005, Vol. 17, No. 4

[7] Jain, Anil K., M. Narasimha Murty, and Patrick J. Flynn. "Data clustering: a review." ACM computing surveys (CSUR) 31.3 (1999): 264-323.

[8] Yadav, Chanchal, Shuliang Wang, and Manoj Kumar. "Algorithm and approaches to handle large Data-A Survey." arXiv preprint arXiv:1307.5437 (2013).

[9] Lawrence 0. Hall, Nitesh Chawla , Kevin W. Bowyer, "Decision Tree Learning on Very Large Data Sets", IEEE, Oct 1998

[10] Mr. D. V. Patil, Prof. Dr. R. S. Bichkar, "A Hybrid Evolutionary Approach To Construct Optimal Decision Trees with Large Data Sets", IEEE, 2006

[11] Guillermo Sinchez-Diaz , Jose Ruiz-Shulcloper, "A Clustering Method for Very Large Mixed Data Sets", IEEE, 2001

[12] Tan, P., Steinbach, M. and Kumar, V. (2005) Cluster Analysis: Basic Concepts and Algorithms. In: Introduction to Data Mining, Addison-Wesley, Boston.

[13] Emily Namey, Greg Guest, Lucy Thairu, Laura Johnson, "Data Reduction Techniques for Large Qualitative Data Sets", 2007

[14] Moshe Looks, Andrew Levine, G. Adam Covington, Ronald P. Loui, John W. Lockwood, Young H. Cho, "Streaming Hierarchical Clustering for Concept Mining", IEEE, 2007

[15] Yen-ling Lu, chin-shyurng fahn, "Hierarchical Artificial Neural Networks For Recognizing High Similar Large Data Sets. ", Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, August 2007

[16] Shuliang Wang, Wenyan Gan, Deyi Li, Deren Li "Data Field For Hierarchical Clustering", International Journal of Data Warehousing and Mining, Dec. 2011

[17] Tatiana V. Karpinets, Byung H.Park, Edward C. Uberbacher, "Analyzing large biological datasets with association network", Nucleic Acids Research, 2012

[18] Bezdek J. C., Ehrlich R., Full W., "FCM: The Fuzzy c-Means Clustering Algorithm," Computers and Geosciences, vol. 10, no. 2-3, p 191-203, 1984.

[19] Al Shami, A., Lotfi, A., Coleman, S. "Intelligent synthetic composite indicators with application", Soft Computing, 17(12), 2349-2364, 2013.

[20] Maimon, O. Z., & Rokach, L. (Eds.). "Data mining and knowledge discovery handbook" Vol. 1. Springer, New York. 2005.

[21] Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Y Zomaya, A., Khalil, I., ... & Bouras, A. "A Survey of Clustering Algorithms for Big Data: Taxonomy & Empirical Analysis." IEEE, 2014.

[22] Guha, S., Rastogi, R., & Shim, K. "CURE: an efficient clustering algorithm for large databases." In ACM SIGMOD Record (Vol. 27, No. 2, pp. 73-84). ACM, 1998.

[23] Moretti, C.; Steinhaeuser, K.; Thain, D.; Chawla, N.V., "Scaling up Classifiers to Cloud Computers," Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on , vol., no., pp.472,481, 15-19 Dec. 2008.

[24] Esteves, R.M.; Pais, R.; Chunming Rong, "K-means Clustering in the Cloud – A Mahout Test," Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on , vol., no., pp.514,519, 22-25 March 2011.