Clustering Big Urban Dataset

Ahmad Al Shami, Weisi Guo, Ganna Pogrebna
Warwick Institute for the Science of Cities, School of Engineering, The University of Warwick
Coventry, CV4 7AL, United Kingdom
Email: a.al-shami@warwick.ac.uk, weisi.guo@warwick.ac.uk

***Abstract*—Cities are producing and collecting massive amount of data from various sources such as transportation network, energy sector, smart homes, tax records, surveys, LIDAR data, mobile phones sensors etc. All of the aforementioned data, when connected via the Internet, fall under the Internet of Things (IoT) category. To use such a large volume of data for potential scientific computing benefits, it is important to store and analyze such amount of urban data using efficient computing resources and algorithms. However, this can be problematic due to many challenges. This article explores some of these challenges and test the performance of two partitional algorithms for clustering Big Urban Datasets, namely: the K-Means vs. the Fuzzy c-Mean (FCM). Clustering Big Urban Data in compact format represents the information of the whole data and this can benefit researchers to deal with this reorganized data much efficiently. Our experiments conclude that FCM outperformed the K-Means when presented with such type of dataset, however the later is lighter on the hardware utilisations.**

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

I. Introduction

The challenges of Big Data are due to the 5Vs which are: Volume, Velocity, Variety, Veracity and Value to be gained from the analysis of Big Data [1]. Many researchers are dealing with different types of data sets, the concern here is to wither to introduce a new algorithm or to use the existing ones to suit large datasets. 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 reduce 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.

II. K-Means vs. Fuzzy c-Means

To highlight the advantages to scientific 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:

1) K-Means: This is a widely used clustering algorithm as it partition a data set into K clusters (C1;C2; ::: ;CK), represented by their arithmetic means called the centroid, which is calculated as the average of all data points (records) belonging to certain cluster.
2) Fuzzy c-Means (FCM) was introduced by [16] and it is derived from the K-means concept for the purpose of clustering datasets, but it differs in that the object may belong to more than one cluster at the same time with a certain degree of belonging to each cluster.
   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. Fuzzy clusters can be characterised by class membership function matrix, and cluster centres are determined first at the learning stage, and then the classification is made by the comparison of Euclidean distance between the incoming features and each cluster centre [17]. 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:

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

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

   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.

$$J(U,V) = \sum_{j=1}^{n} \sum_{i=1}^{c} (u_{i,j})^2 \left\| x_j - v_i \right\| \tag{3}$$

   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.

a) Estimate the class membership U.

b) Calculate vectors of cluster centres $V = \{v_1, v_2, \ldots, v_i, \ldots v_c\}$ using the following expression:

$$v_i = \frac{\sum_{j=1}^{n} (u_{i,j})^2 x_j}{\sum_{j=1}^{n} (u_{i,j})^2} \qquad i = 1, 2, \ldots, c \tag{4}$$

c) Update the class membership matrix U with:

$$u_{i,j} = \frac{1}{\sum_{r=1}^{c} \left(\frac{\|x_j - v_i\|}{\|x_j - v_r\|} \right)^2} \quad i = 1, \ldots, c; \quad j = 1, \ldots, n \tag{5}$$

d) 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 [17].

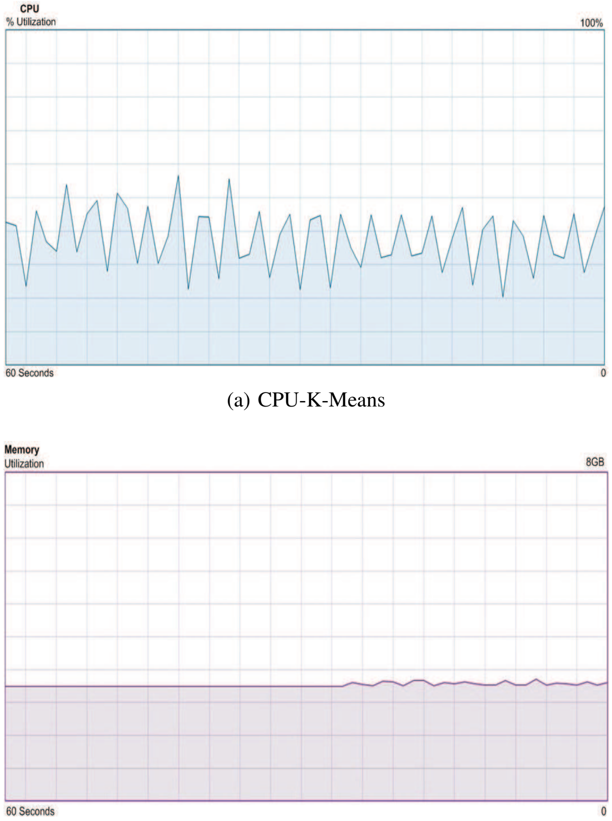
III. Experiments Setup

Lidar dataset were used for this experiment as it is gaining importance for urban planning such as floodplain mapping, hydrology, geomorphology, landscape ecology, coastal engineering, survey assessments, and volumetric calculations. The experiments carried to compare how the candidate K-Means and FCM clustering techniques cope with clustering big urban dataset using mid-range level 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 operating system. The algorithms were implemented against a a LIDAR data points [3], 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.

IV. Comparative Analysis

1) K-Means Clustering: This clustering technique is applied to the specified dataset starting with a small cluster number K = 5 and gradually increased to K = 25 clusters. Fig. 1 shows how on average the used hardware fared to obtain the desired number of K clusters and Table I lists a summary of the statistics of elapsed time and resources used for K-Means algorithm to converge.

2) FCM Clustering: This clustering technique was also applied to the same generated dataset with cluster number starting with 5 and gradually increased to reach 25 clusters. Fig. 2-a and Fig. 2-b show the CPU and RAM usage while executing the large dataset with FCM clustering function and Table II lists summary of the main time and resources it took the FCM algorithm to converge for the different number of assigned clusters.



(a) CPU-K-Means

(b) RAM-K-Means

Fig. 1: Average CPU and Memory usage during K-Means execution.(a) CPU, (b) RAM.

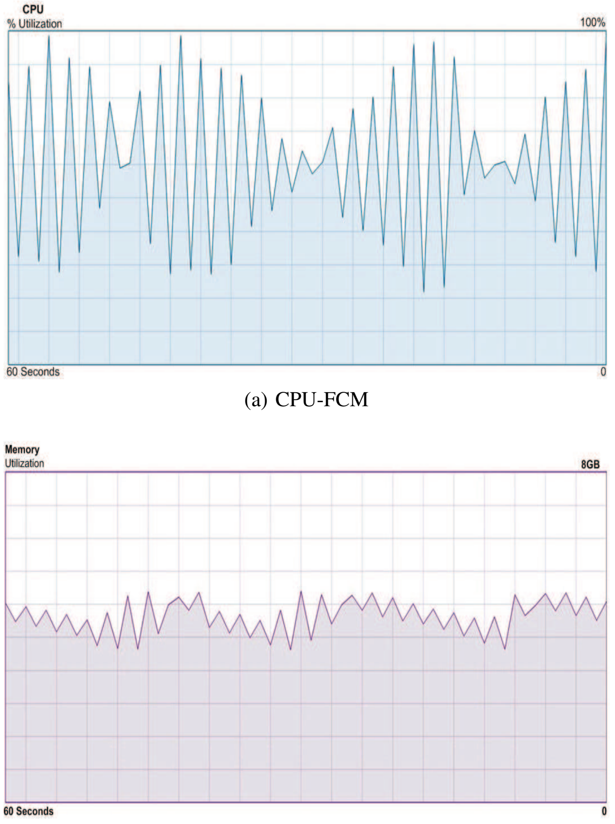
TABLE I: Time elapsed and resources used for K-Means clustering.

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%**

By comparing the results in Table I and Table II, it is clear that the lowest average time measured for FCM to regroup the data was 127.520 seconds, while it took K-Means an average of 327.558 seconds to form the same number of clusters. On



(a) CPU-FCM

(b) RAM-FCM

Fig. 2: Average CPU and Memory usage during FCM execution.(a) CPU, (b) RAM.

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 cores 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.

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 [18]. Also, this fuzzy partitioning technique and due to its fuzzy capability, FCM could produce a better quality of the clustering output [19] which could benefit many data analysts.

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 another types of big datasets.

References

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

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

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

[4] 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

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

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

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

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

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

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

[11] Emily Namey, Greg Guest, Lucy Thairu, Laura Johnson, “Data Reduction Techniques for Large Qualitative Data Sets”, 2007

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

[13] 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

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

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

[16] 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.

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

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.