Optimal K-Means Clustering with Dual Distance Cost Estimation

Silky Chourasia[1], Brij Kishore[2]
[1]*Student, Dept. of Computer Science and Engg., Apex Institute of Engineering and Technology, Jaipur, India*
[2]*Assistant Professor, Dept. of Computer Science and Engg., Apex Inst. of Engg. and Technology, Jaipur, India*

***Abstract*: This paper is intended to present an algorithm for the K-means Clustering which can identify the nearest cluster as well as the best cost cluster form is required too. The developed algorithm is distance based algorithm. K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity. The results of the K-means clustering algorithm are: The centroids of the K clusters, which can be used to label new data Labels for the training data (each data point is assigned to a single cluster).**

***Keywords*: K-means, Data Mining, Euclidean Distance, Manhattan Distance Formula, Iterative Active Clustering.**

1. Introduction

The K-Means clustering algorithm is used to find groups which have not been explicitly labeled in the data. This can be used to confirm business assumptions about what types of groups exist or to identify unknown groups in complex data sets. Once the algorithm has been run and the groups are defined, any new data can be easily assigned to the correct group.

Clustering is a way that classify the raw data reasonably and searches the hidden patterns that may exist in datasets. It is a process of grouping data objects into disjointed clusters so that the data in the same cluster are similar, yet data belonging to different cluster differ. The demand for organizing the sharp increasing data and learning valuable information from data, which makes clustering techniques are widely applied in many application areas such as artificial intelligence, biology, customer relationship management, data compression, data mining, information retrieval, image processing, machine learning, marketing, medicine, pattern recognition, psychology, statistics and so on.

K-means is a numerical, unsupervised, non-deterministic, iterative method. It is simple and very fast, so in many practical applications, the method is proved to be a very effective way that can produce good clustering results. But it is very suitable for producing globular clusters. Several attempts were made by researchers to improve efficiency of the k-means algorithms. In previous researches, there is an improved k-means algorithm based on weights. This is a new partitioning clustering algorithm, which can handle the data of numerical attribute, and it also can handle the data of symbol attribute. Meanwhile, this method reduces the impact of isolated points and the “noise”, so it enhances the efficiency of clustering. However, this method has no improvement on the complexity of time. In previous researches, it proposed a systematic method to find the initial cluster centers, this centers obtained by this method are consistent with the distribution of data. Hence this method can produce more accurate clustering results than the standard k-means algorithm, but this method does not have any improvements on the executive time and the time complexity of algorithm. This paper presents an improved k-means algorithm. Although this algorithm can generate the same clustering results as that of the standard k-means algorithm, the algorithm of this paper proposed is superior to the standard k-means method on running time and accuracy, thus enhancing the speed of clustering and improving the time complexity of algorithm. By comparing the experimental results of the standard k-means and the improved k-means, it shows that the improved method can effectively shorten the running time.

2. Literature survey

Many methods have been proposed for clustering on dynamic or incremental datasets such as recently developed density based algorithms [1]. In these type of methods, an updated list of cluster densities must be generated after applying new changes. Even though this can be done in linear time, it is not optimal because these algorithms can’t use the previously calculated data without approximations [2], which means that we must do all of the calculations again in order to generate fully accurate clusters. Most of the current methods don’t use their own previous results as a source of data for clustering the dataset after submitting the changes because they’re not designed for reclustering (Applying the new changes dynamically and update the clusters). Some modifications must be done on the basis of these algorithms to let them use the differential data for updating the results instead of recalculating them [3].

One of the main types of clustering algorithms that is fairly commonly used is centroid based algorithms [4], out of which

K-means is the most commonly-used. The basic K-means method was proposed by Lloyd in 1957 [5]. Since then, many researchers worked to enhance K-means Clustering algorithm for different purposes. Some tried to accelerate the algorithm by reducing the redundant centroid calculations with mathematical approaches, for example by using triangle inequality [6] or by using KD-trees [7] or blacklisting methods [8]. Elkan's proposed method [9], defines an upper bound for each node based on its distance to its current cluster centroid. It reduces the calculations by skipping the clusters where the node can't move on to, which accelerates K-means eventually. Hamerly proposed an- other similar algorithm [10] which was using the second nearest centroid for each node to determine a lower bound for them as well. Later, Drake and Hamerly proposed a better method for accelerating K-means by using adaptive distance bounds [11]. Another approach for accelerating K-means was filtering and ignoring the nodes that are not going to move to a new cluster anymore [12]. These methods use an extra dataset to keep track of the closer nodes to centroids. It's not possible to determine which version will work faster on a certain dataset before testing it, as the initializing factors (how first centroids were chosen and the number of clusters - K) and nodes density (the structure of dataset) have direct effects on these methods efficiency. However, the presented test results show that Hamerly enhancements are doing generally better on some typical datasets [13].

Another type of proposed enhancements on K-means was trying to increase the algorithm accuracy [14]. For that, one needs to define the meaning of accuracy as "clustering is in the eyes of beholder" [15] meaning one's definition of a "correct" clustering on a specific dataset depends on what they expect from the clustering system.

Some others were focused on implementing K-means on data with specific requirements. For example, some requirements are using it on data with more dimensions [16] or using it on peer to peer networks and mobile networks [17]. As we can't always have a server to do the calculations, these enhancements rely on their predictions and local synchronization to achieve an acceptable result. These algorithms are trying to converge on a set of centroids that are as close as possible to the centroids that would have been produced if the data from all the nodes were first centralized then K-means were run [16]. In other words, they use a nearly accurate cluster map to cover the peer to peer relationship between the separated nodes in the graph. Recently, a lot of developments, protocols, and algorithms have been proposed that are following the same trend [17].

Furthermore, to be able to use clustering or re-clustering algorithms in practice (for example in image processing and computer vision [18] or for Routing [19] some modifications had been done on them. These algorithms are enormously better than the theoretical older and original versions for their specific applications.

Another enhancement was improving K-means to be able to keep producing results in a dynamic network [20]. When we use the term "Dynamic Network" we refer to a network that is changing over time. Most of these enhanced algorithms, will only apply the change on the cluster which had the node originally (if the change is a node removal) or the one which is getting this new node (if the change is a node addition) and will ignore the impact of that change on other clusters because they want to re-cluster dynamically [21]. The only update these algorithms will do is to notify all the clusters about the centroid movements to keep the whole clustering system valid, but the result is not optimal. Since the movement impact on neighbor clusters was ignored, there might be nodes in that neighborhood which are not in a cluster with the nearest centroid [22].

The amount of the nodes that are not optimally clustered will aggrandize over time, which will increase the clustering error after a long stream of changes are applied on the network [23]. That is why making sure the algorithm will stay valid overtime was another challenge in this topic. Several methods for working on dynamic incremental networks [24] have been purposed. The core idea of all of these approaches is to use the last valid clustered network and the set of centroids of that network to re-initialize the original algorithm on the whole network.

Re-clustering and determining the best cluster for the nodes in an online data stream is the state of the art. The suggested algorithms will work on any cluster shape but they can only work on inserted nodes and won't consider node removals so these algorithms can't consider the changing in the location of nodes or node links [25]. In addition, we can use the mathematical basis of the still fairly used centroid based clustering algorithms to achieve a better result.

3. Proposed algorithm

K-means classical method works over the fix cluster count, however in this paper an algorithm with multiple cluster count and hence an optimal cluster is picked from all the generated clusters. A dataset of any length can be taken as an input, source of it here in this thesis is UC Irvine. Also the length of the dataset for the test purposes varies from 20 to 1000 depending upon the availability of the system hardware.

The algorithm works over the principle of selecting the optimal cluster out of the numerous cluster on the basis of the shortest distance between the centroids and the participant points of the workspace selected for the clustering.

Proposed methodology works over creating 9 cluster starting with 2 clusters and maximum it goes up to 10 clusters in which the participating data of the dataset gets clustered, for selecting the centroid a MATLAB script is designed which on the basis of number of clusters required selects the centroid from the dataset.

On further processing the algorithm uses the Euclidean and the Manhattan Distance formula for calculating the distance between the centroids and the rest of the dataset values or coordinates.

The Euclidean distance or Euclidean metric is the "ordinary" straight-line distance between two points in Euclidean space. With this distance, Euclidean space becomes a metric space. The associated norm is called the Euclidean norm. Older literature refers to the metric as the Pythagorean metric.

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

The Manhattan distance function computes the distance that would be traveled to get from one data point to the other if a grid-like path is followed. The Manhattan distance between two items is the sum of the differences of their corresponding components.

$$d_1(\mathbf{p},\mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

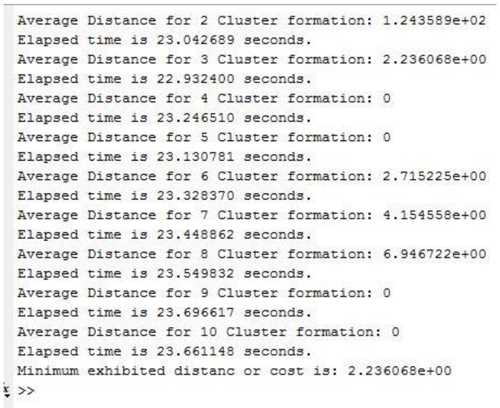
On the successful cluster creation algorithm will calculate the shortest median distance amongst all the clusters either by the Euclidean distance formula or by the Manhattan distance formula.

4. Result

Proposed K-Means method omits the dilemma of selecting the most optimal cluster formed amongst the n number of clusters also it saves a lot of data mining which may require more memory and more hardware for its processing.

The presented figure shows the generated cluster distances and finds the best cost amongst all the clusters.

Figure 2 shows the sample cluster formed over a 2 plot of MATLAB test environment.



```
Average Distance for 2 Cluster formation: 1.243589e+02
Elapsed time is 23.042689 seconds.
Average Distance for 3 Cluster formation: 2.236068e+00
Elapsed time is 22.932400 seconds.
Average Distance for 4 Cluster formation: 0
Elapsed time is 23.246510 seconds.
Average Distance for 5 Cluster formation: 0
Elapsed time is 23.130781 seconds.
Average Distance for 6 Cluster formation: 2.715225e+00
Elapsed time is 23.328370 seconds.
Average Distance for 7 Cluster formation: 4.154558e+00
Elapsed time is 23.448862 seconds.
Average Distance for 8 Cluster formation: 6.946722e+00
Elapsed time is 23.549832 seconds.
Average Distance for 9 Cluster formation: 0
Elapsed time is 23.696617 seconds.
Average Distance for 10 Cluster formation: 0
Elapsed time is 23.661148 seconds.
Minimum exhibited distanc or cost is: 2.236068e+00
>>
```

Fig. 1. Best Cost Calculations

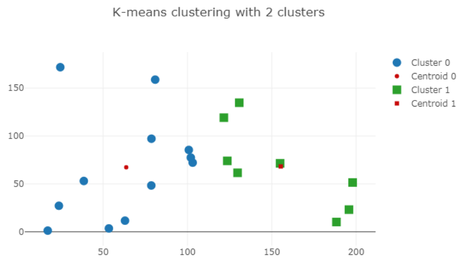


Fig. 2. Cluster Plot

5. Conclusion

This paper presented an overview on optimal K-Means clustering with dual distance cost estimation

References

[1] Aaron, B., Tamir, D. E., Rishe, N. D., & Kandel, A. (2014). Dynamic incremental K-means clustering. In Proceedings of computational science and computational intelligence (CSCI): 1 (pp. 308–313).

[2] Aggarwal, C. C. (2013). A survey of stream clustering algorithms. In Data clustering algorithms and applications (pp. 231–258). Taylor & Francis.

[3] Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., & Park, J. S. (1999). Fast algorithms for projected clustering. ACM SIGMoD Record, 28 (2), 61–72.

[4] Kriegel, H. P., Kroger, P., Sander, J., & Zimek, A. (2011). Density-based clustering. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1 (3), 231–240.

[5] Krishna, K., & Murty, M. N. (1999). "Genetic K-means algorithm. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 29 (3), 433–439.

[6] Likas, A., Vlassis, N., & Verbeek, J. J. (2003). "The global k-means clustering algo- rithm. Pattern recognition, 36 (2), 451–461.

[7] Lin, C. R., & Gerla, M. (1997). Adaptive clustering for mobile wireless networks. IEEE Journal on Selected areas in Communications, 15 (7), 1265–1275.

[8] Lloyd, S. (1982). Least squares quantization in PCM. IEEE Transactions on Information Theory, 28 (2), 129–137.

[9] Lua, E. K., Crowcroft, J., Pias, M., Sharma, R., & Lim, S. (2005). A survey and compar- ison of peer-to-peer overlay network schemes. IEEE Communications Surveys & Tutorials, 7 (2), 72–93.

[10] MacKay, D. (2003). Chapter 20. An example inference task: Clustering" (PDF). In Information Theory, inference and learning algorithms (pp. 284–292). Cambridge University Press.

[11] Mobasher, B., Cooley, R., & Srivastava, J. (1999). "Creating adaptive web sites through usage-based clustering of URLs. In Proceedings of knowledge and data engineering exchange, 1999. (KDEX'99) (pp. 19–26).

[12] Mohammed, R. S., Abbood, F. H., & Yousif, I. A. (2016). "Image encryption technique using clustering and stochastic standard map. In Proceedings of multidisciplinary in IT and communication science and applications (AIC-MITCSA (pp. 1–6).

[13] Nidoy, E. W. k-MACE clustering for gaussian Gaussian clusters.

[14] Nguyen, J., Thuy, T. T., & Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. IEEE Communications Surveys & Tu- torials, 10 (4), 56–76.

[15] Papadopoulos, S., Kompatsiaris, Y., Vakali, A., & Spyridonos, P. (2012). "Community detection in social media. Data Mining and Knowledge Discovery, 24 (3), 515–554.

[16] Pelleg, D., & Moore, A. (1999). Accelerating exact k-means algorithms with geometric reasoning. In Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining (pp. 277–281).

[17] Pena, J. M., Lozano, J. A., & Larranaga, P. (1999). An empirical comparison of four initialization methods for the k-means algorithm. Pattern Recognition Letters, 20 (10), 1027–1040.

[18] Pham, D. T., Dimov, S. S., & Nguyen, C. D. (2004). An Incremental K-means algorithm. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 218 (7), 783–795.

[19] Phillips, S. J. (2002). "Acceleration of k-means and related clustering algorithms. In Proceedings of workshop on algorithm engineering and experimentation: (pp. 166–177).

[20] Piatetsky-Shapiro, G. (1996). In Usama M. Fayyad, Padhraic Smyth, & Ramasamy Uthurusamy (Eds.). Advances in knowledge discovery and data mining: 21.

[21] Menlo Park: AAAI press. Portnoy, L., Eskin, E., & Stolfo, S. (2001). "Intrusion detection with unlabeled data using clustering. In Proceedings of ACM CSS workshop on data mining applied to security (DMSA-2001).

[22] Pukhrambam, P., Bhattacharjee, S., & Das, H. S. (2017). A multi-level weight based routing algorithm for prolonging network lifetime in cluster

based sensor net- works. in Proceedings of the international conference on signal, networks, computing, and systems (pp. 193–203).

[23] Reddy, R., & Chaitany, V. V. K. (2016). Reducing current availability and re-clustering time in sensor nets. IJITR, 4 (6), 4656–4658.

[24] Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (1998). Density-based clustering in spatial databases: The algorithm gdbscan and its applications. Data Mining and Knowl- edge Discovery, 2 (2), 169–194.

[25] Sculley, D. (2010). "Web-scale k-means clustering. In Proceedings of the 19th ACM international conference on World wide web (pp. 1177–1178).