DOI: 10.24996/ijs.2018.59.1A.18

Iraqi Journal of Science

ISSN: 0067-2904

Proposed KDBSCAN Algorithm for Clustering

Yossra Hussein, Safa Abdel Jalil*
Department of Computer Science, College of Science, Technology University, Baghdad, Iraq.

Abstract

Science, technology and many other fields are use clustering algorithm widely for many applications, this paper presents a new hybrid algorithm called KDBSCAN that work on improving k-mean algorithm and solve two of its problems, the first problem is number of cluster, when it`s must be entered by user, this problem solved by using DBSCAN algorithm for estimating number of cluster, and the second problem is randomly initial centroid problem that has been dealt with by choosing the centroid in steady method and removing randomly choosing for a better results, this work used DUC 2002 dataset to obtain the results of KDBSCAN algorithm, it`s work in many application fields such as electronics libraries, biology and marketing, the KDBSCAN algorithm that described in this paper has better results than traditional K-mean and DBSCAN algorithms in many aspects, its preform stable result with lower entropy.

Keywords: clustering, K-mean, DBSCAN, KDBSCAN.

:

خوارزمية KDBSCANالمقترحة للتجميع

يسرى حسين، صفا عبد الجليل*

قسم الحاسبات، كلية العلوم، الجامعةالتكنولوجية، بغداد، العراق.

الخلاصة

العلوم والتكنولوجيا والعديد من المجالات الاخرى تستخدم خوارزميات التجميع بصورة كبيرة للعديد من التطبيقات، هذ البحث يقدم خوارزمية دمج جديدة تسمى KDBSCAN والتي تعمل على تطوير خوارزمية K-mean لحل اثنان من مشاكلها، المشكلة الاولى هو عدد المجاميع، والذي يجب ان يتم ادخاله عن طريق المستخدم، وتم حل هذه المشكلة عن طريق استخدام خوارزمية DBSCAN لتخمين عدد المجاميع، و المشكلة الثانية هو الاختيار العشوائي للمراكز، و الذي تم تعامل معهاعن طريق المراكز بطريقة ثابتة وازالة عشوائية الاختيار للحصول على نتائج افضل، تم العمل بأستخدام قاعدة بيانات DUC 2002 للحصول على النتائج خوارزمية KBSCAN، هي تعمل في تطبيقات متعددة مثلا المكتبات الالكترونية، علم البايولوجي و مراكز التسوق، خوارزمية KDBSCAN التي تم وصفها في هذا البحث لها نتائج احسن من الخوارزميتين K-mean التقليدية و DBSCAN في جوانب متعددة، حيث انها توفر نتائج ثابتة وعشوائية قليلة.

1. Introduction

Clustering techniques are used to partition dataset into several groups each group has objects shear similar features, while has different feature from other groups, clusters represents by those groups, clustering is unsupervised method works to learn the unlabeled data, the main purpose of all clustering methods is to offer a perfect partition for set of objects also it offers a prediction to group the ungrouped data [1], clustering algorithms has wild rang in world its use publicly in many application such as pattern recognition, artificial intelligence, information technology, image

*Email: safa.abdel.jalil@gmail.com

processing, biology such as Bioinformatics (gene expression analysis), psychology, information retrieval, and marketing [2]. This work hybrid two algorithms K-mean and DBSCAN to create KDBSCAN algorithm where K-Mean algorithm is one of most popular algorithm of partition based method that used for many purpose but still has many disadvantages that needs to improve [3], while DBSCAN is one of Density based method which its work with database according to its density of the neighborhood items [4].

2. Relation Work

In traditional k-mean the distance between centroid and dataset points are calculated in all iterations, in 2010 S. Na et. al.[5] proposed to have the distance from new centers only once, if the distance is equal or less than previously calculated distance then it remains in cluster itself, else it will run for same procedure until all data points are assigned to closest centroids, this improve make the proposed system much faster than traditional K-mean algorithm.

juntao wang et. al. in 2011 [6] improve one of major k-mean clustering algorithm drawbacks, this drawbacks is k-mean weakness for outlier and noise data, so the authors enhanced this weak point by using noise data filter, "density based outlier detection method" is used on dataset to be clustered and remove noise and outliers, the proposed system has less clustering time and more accuracy but if the data set of proposed system is huge it will cost more time.

Ghousia Usman et.al. in 2013 [7] proposed system for choosing initial centroids in k-mean algorithm since initial centroid randomly choosing is one of important k-mean drawbacks, so they proposed effective methods summarize in three steps 1) use Euclidean measurement to measure the distance between data point 2) find the nearest data point which are similar 3) those similar data point will be the actual centroids, the proposed system improve k-mean in accuracy and effectiveness.

Vighnesh and Damodar in 2014 [8] enhanced K-mean clustering algorithm by removing two of its problem number of cluster and randomly initial centroids, the enhanced applied by using fast heuristic algorithm for estimate number of cluster as well as initial centroids, the proposed system show better results in effectiveness unlike the traditional K-mean.

In 2015 Bouhmala et. al. [9] try to improve the speed of K-mean, so they combine K-mean with genetic algorithm create "GAKM", this combination is tested over several datasets like iris and glass, GAKM has two phases first apply genetic search algorithm on database to create clusters using two-point crossover then second phase apply k-mean algorithm technique to improve the accuracy and quality of the created cluster ,the results of GAKM is faster than K-mean and genetic algorithms but it can`t get the best clusters.

Arpit Bansal et.al. in 2017 [10] proposed a system based on k-mean algorithm to deal with two of k-mean problem which is accuracy level and execution time, the proposed system enhance k-mean algorithm and enhanced Euclidian measure, the enhanced Euclidian distance based on find normal distance by using normalization and enhanced k-mean cluster database based on majority voting, the proposed system has double accuracy level then traditional and with very little time.

3. K-mean Algorithm

K-mean algorithm is data mining technique for clustering data but its need number of cluster as input and find initial centroids randomly which make the result unstable with each single execution, but K-mean algorithm is simplicity and fast make it mostly used, it determines number of cluster by user as K then generate K point as initial centers randomly, for each cluster one center, then each point are assign to the closest center by using distance measure [11], Euclidean distance is mostly use with K-mean algorithm is usually determine the distance between two point and center [12]. The Euclidean distance between multi-dimensional data points

X = (x1, x2, x3... xn) and

Y = (y1, y2, y3... yn) is described as follows:

$$d(X,Y)=\sqrt{(x_1 - y_2)^2 + (x_2 - y_2)^2 + .. + (x_m - y_m)^2} \quad \text{.....(1)}$$

Then the center of each cluster update by finding the mean value of each cluster, sometimes points change from cluster to cluster the approach end when no changing happened [13].

Algorithm 1: K-mean clustering Algorithm [13]

1) Randomly select k data object from dataset D as initial cluster centers.
2) Repeat
- Calculate the distance between each data object di (1<=i<=n) and all k cluster centers j (1<=j<=n) and assign data object di to the nearest cluster.
- For each cluster j (1<=j<=k), recalculate the cluster center.
- Until no changing in the center of clusters.

K-Mean clustering algorithm is faster than hierarchical clustering when there is large number of variables, also its produce tighter clusters if the clusters are globular, but its drawbacks are breaks in several points, firstly there are difficult in clustering quality comparing for randomly initial centroids, secondly number of cluster must be input by user also it's hard to estimate, thirdly and lastly K-Mean is not deal with noise and outliers [14], this paper solve some of this drawbacks.

4. DBSCAN Algorithm

DBSCAN is one of effective density-based cluster algorithm, its simple and create cluster in arbitrary shape also cluster information as noise and outliers, DBSCAN is fast and its efficient with large Data Base, it has two inputs first radius second the minimum points found in radius, DBSCAN general idea based on finding minimum number of point in radius space to form clusters else if the number of minimum points not found in radius it will mark as noise [15] [3], DBSCAN algorithm is stated below [16] .

Algorithm 2: DBSCAN outlier [16]

- Set of points to be considered to form a graph.
- Create an edge from each point c to the other point in the neighborhood of c.
- If set of nodes N not contain any core points then terminate N.
- Select a node X that must be reached form c.
- Repeat the procedure until all core points forms a cluster.

5. The KDBSCAN Algorithm Architecture

K-mean algorithm is used widely but also it has number of disadvantage, one of those disadvantage is number of cluster must be estimate and produce as input for K-mean algorithm also selecting initial centers arbitrarily that led to different results in each program-run [17].

The KDBSCAN algorithm works on two parts: first part applies preprocessing operations to the database which are three phases tokenizing, stop words removal and stemming, tokenizing performs on every word in the document and put it in a token to be handled easier and faster, stop word removal removes the unwanted words that are unimportant for clustering the dataset these words like is, are, an, un.. etc to speed up the system execution.

The stemming process is done by reducing words to their root form by removing the changes between several forms for the same word as example computer, computing they back to their root compute, stemming process in addition remove the difference between lowercase and uppercase of the words. There are many stemming algorithm, this work depends on a manual method as one of stemming operations to produce word`s root.

The first step of part two in KDBSCAN algorithm uses the result from preprocess operation in part one as an input to the DBSCAN algorithm for scanning the processed dataset and found the true approximate number of cluster, the output of DBSCAN algorithm will be the input for K-mean algorithm which is the second step of part two in KDBSCAN algorithm that based on selecting initial center in particular method rather than using random choosing where the dataset breaks into blocks equal to number of cluster, then find each block median with index (i) and set it as center then compare the current center with pervious center if the two centers are close together then center with index (i) will remove and replaced with index point (i-1), algorithm 3 describes the KDBSCAN algorithm.

Algorithm 3: KDBSCAN algorithm

Input: D Dataset containing p words, K=0, eps: the radius parameter, and *MinPts*: the neighborhood density threshold
Output: clusters of dataset

Process:
Step 1: Preprocess
Tokenization / Separate data in tokens
Remove stop words process / remove the unimportant words like is, un, the.
Stemming process / reducing words to their root form by using manual method which are based on regular expression.
Step 2: for each unvisited p in D do
Step 2.1: for each p` in D do
distance = levenshtein_distance (p, p`, eps)

if distance < eps then
p_neighbor = p_neighbor+1
end if
next

step 2.2: if p_neighbor < minimum points then
marke p as noise
else
k= new cluster
for each p`` in p`_eps do
distance = levenshtein_distance (p, p`, eps)
if distance < eps then
p_neighbor = p_neighbor+1
end if
next
end if
next
Step 3: Divide the D into k subsets
Find CENTERS by
For i=0 to k
If i=0 then
Cluster_center(i) = median of first subset.
Else
Cluster_Center(i) = median_subset(i) with index x in D // find median of subset(i)
For j=0 to i
A= | cluster_center(i) – cluster_center(j) |
If (A > 3) then
cluster_center(i)=index(x-1)
end if
End for
End if
End for
Step 4: repeat
Re-distribute the objects to the closest cluster according to its distance to the cluster center / Using levenshtein distance function
Until no change
End.

The above algorithm describe the KDBSCAN algorithm in details, it's important to notice in step 3 parameter A is measuring the difference between cluster_center(i) and cluster_cnter(j) and if the

difference is more than 3 then the center will shift to index (x-1), this operation is for reducing the similarity between centers, number 3 is used as threshold and it gives the best results, in step 4 levenshtein distance is used for matching two words and gives the different between theme since Dataset D is a collection of words.

6. Experimental results

This paper present a proposed algorithm use DBSCAN to estimate number of cluster, and remove randomly initial centroids chose and make result stable also improve accuracy by reduce the entropy compering with classical K-mean, DUC 2002 dataset was used as a source of words by assembling 2600 words, Table-1 described the total results so as Figure-1, KDBSCAN algorithm was written in vb.net language version 2015 on widows10.

Table 1-Comparison between classical K-mean, DBSCAN and KDBSCAN

Number of cluster	Classical K-mean		DBSCAN		KDBSCAN	
	Entropy	Run time in Seconds	Entropy	Run time in Seconds	Entropy	Run time in Seconds
2	0.361	2.39	0.3	12.3	0.127	2.32
3	0.298	2.31	0.454	11.7	0.180	2.3
4	0.529	2.33	0.4	12.32	0.093	2.31
5	0.116	2.41	0.390	12.23	0.109	2.4
6	0.117	2.45	0.385	12.35	0.061	2.44
7	0.186	2.46	0.348	12.06	0.088	2.45
8	0.127	2.47	0.440	11.7	0.104	2.45
9	0.530	2.54	0.370	12.31	0.109	2.51
10	0.164	2.7	0.380	12.36	0.088	2.61

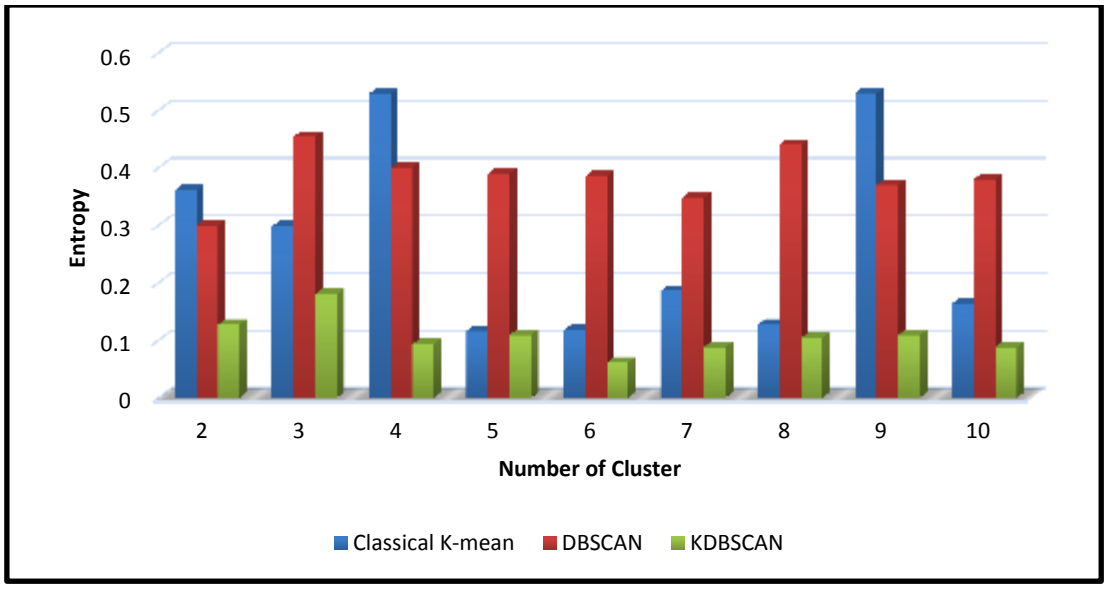

Figure1- Comparison between classical K-mean, DBSCAN and KDBSCAN

This work depends on entropy values to evaluate algorithms performers results, entropy measures the randomness in data clustering, so the higher value of entropy the more randomness in clustering, from Table-1, it`s obvious that KDBSCAN algorithm has the lowest entropy values in all cases compared with its counterparts K-mean and DBSCAN algorithms, that means KDBSCAN algorithm clusters data more correctly and tend to be optimal, because the highest entropy value recorded is 0.180, while K-mean algorithm is 0.530 and DBSCAN algorithm is 0.454. From Figure-1 it`s obviously that KDBSCAN algorithm entropy values are similar in all cases; that raged between 0.06-0.180, then DBSCAN algorithm that entropy values are range between 0.3- 0.440, as for classical k-mean there is a great difference between its entropy values because of random center selection.

7. Conclusion

This paper introduce a new algorithm KDBSCAN that improve K-mean algorithm by using DBSCAN algorithm to find number of cluster as to be an input for k-mean algorithm, and in depend

on number of cluster k-mean breaks the dataset into parts and choose the median item to be one of initial centroids if this item not similar to other centroids, from Table-1 the difference between KDBSCAN, traditional k-mean and DBSCAN in entropy values is large for example when number of cluster 4 entropy for traditional k-mean is 0.529 with run time equal to 2.33 second and in DBSCAN entropy is 0.4 with run time equal to 12.32 second, while in KDBSCAN entropy value equal to 0.093 and run time is 2.31 second, the different in entropy between KDBSCAN and traditional k-mean is 0.436, and between KDBSCAN and DBSCAN in entropy is 0.307, when number of cluster 2,3,7,9 there is the same large different in entropy value, two of most important k-mean drawback has been solved, estimate number of cluster as well as remove randomly choosing of initial centroids leads to improve accuracy Significantly by reducing entropy.

References

1. Singh, A. A., Fernando, A. E. and Leavline, J. E. **2016.** Performance Analysis on Clustering Approaches for Gene Expression Data. *International Journal of Advanced Research in Computer and Communication Engineering*, **5**(2): 196-200.
2. Gan, G., Ma, C., and Wu, J. **2007**. *Data Clustering Theory, Algorithms, and Applications*. American Statistical Association and the Society for Industrial and Applied Mathematics, E-book
3. Raval, R. U., and Jani C. **2015**. Implementing and Improvisation of K-means Clustering. *International Journal of Computer Science and Mobile Computing*, **4**(11): 72 – 76.
4. Ma, L., Gu, L., Li, B., Qiao, S., and Wang, J. **2015**. MRG-DBSCAN: An Improved DBSCAN Clustering Method Based on Map Reduce and Grid. *International Journal of Database Theory and Application*, **8**(2): 119-128.
5. Na S., Xumin, L., and Yong G. **2010.** Research on k-means clustering algorithms. *IEEE Computer society*, **74**: 63-67.
6. Wang, J., and Su, X. **2011.** An improved K-Means clustering algorithm. Conference on Communication Software and Networks, Xi'an, China, 44 – 46.
7. Usman, G., Ahmad, U., and Ahmad, M. **2013**. Improved K-Means Clustering Algorithm by Getting Initial Cenroids. *World Applied Sciences Journal*, **27**(4): 543-551.
8. Birodkar, V., and Edla, D. R. **2014.** Enhanced *K*-Means Clustering Algorithm using A Heuristic Approach. *Journal of Information and Computing Science*, **9**(4): 277-284.
9. Bouhmala, A. Viken, J. B. Lonnum, **2015.** Enhanced Genetic Algorithm with K-Means for the Clustering Problem. *International Journal of Modeling and Optimization*, **5**(2): 150-154.
10. Bansal, A., Sharma, M., and Goel, S. **2017**. Improved K-mean Clustering Algorithm for Prediction Analysis using Classification Technique in Data Mining. *International Journal of Computer Applications*, **157**(6): 0975 – 8887.
11. Farhan, A. K., AbdulMajeed, G. H., and Ali, R. S. **2015.** ICM Compression System Depending On Feature Extraction. *International Journal of Emerging Trends & Technology in Computer Science*, **4**(3): 47-55.
12. Abdul Nazeer, K. A., and Sebastian, M. P. **2009.** Improving the Accuracy and Efficiency of the k-means Clustering Algorithm. *Proceedings of the World Congress on Engineering*, U.K., London, 23, 1-3.
13. Yedla, M., Pathakota S. R. and Srinivasa, T. M. **2010**. Enhancing K-means Clustering Algorithm with Improved Initial Center. *International Journal of Computer Science and Information Technologies*, **1**(2): 121-125.
14. Zafar, M. H., and Ilyas, M. **2015**. A Clustering Based Study of Classification Algorithms. *International Journal of Database Theory and Application*, **8**(1): 11-22
15. Bäcklund, H., Hedblom, A., and Neijman, N. **2011**. DBSCAN A Density-Based Spatial Clustering of Application with Noise. *Linköpings Universitet – ITN*, http://staffwww.itn.liu.se/aidvi/courses/06/dm/Seminars2011.
16. Sajana, T., Rani,C. M., and Narayana, K. V. **2016.** A Survey on Clustering Techniques for Big Data Mining. *Indian Journal of Science and Technology*, **9**(3).
17. Raval, U. R., and Jani, C. **2016.** Implementing & Improvisation of K-means Clustering Algorithm. *Journal of Computer Science and Information Technology*, **5**(5): 191 – 203.