***G-Means Clustering***

Finding the ‘K’ in K-means clustering

Deepak Patil *(Author)*

Dept. of Computer Science

North Carolina State University

Raleigh, NC, USA

e-mail: dpatil@ncsu.edu

Rasika Pande *(Author)*

Dept. of Computer Science

North Carolina State University

Raleigh, NC, USA

e-mail: rhpande@ncsu.edu

Atefeh Morsali *(Author)*

Dept. of Computer Science

North Carolina State University

Raleigh, NC, USA

e-mail: matefah@ncsu.edu

Satish Gurav *(Author)*

Dept. of Computer Science

North Carolina State University

Raleigh, NC, USA

e-mail: sjgurav@ncsu.edu

Uddhav Bhosle *(Author)*

Dept. of Computer Science

North Carolina State University

Raleigh, NC, USA

e-mail: ubhosle@ncsu.edu

Kumar Nielarshi *(Author)*

Dept. of Computer Science

North Carolina State University

Raleigh, NC, USA

e-mail: knielar@ncsu.edu

*Abstract*—*Most clustering algorithms require the user to specify the number of clusters (called k), and it is not always clear what is the best value for k. G-means algorithm is one of the proposed method by Hamerly and Elkan [2] for learning k while clustering. In this project, the G-means algorithm proposed by Hamerly and Elkan [2], was examined. The tests were run on two datasets, a synthetic dataset [7] and e-coli dataset [8]. In our test, the value of k found was always close to the true number of clusters. The accuracy of G-Means is also examined by comparing its clustering results with the original cluster labels of the datasets. The result of the analysis shows that the accuracy of the G-means algorithm is above .. . this indicate the G-means algorithm performs very well.*

Keywords-G-means, Anderson Darling test, K-means, clustering, Gaussian distribution

# Introduction

Clustering problems arise in many different applications, such as data mining and knowledge discovery, data compression and vector quantization, and pattern recognition and pattern classification [1]. Clustering in data mining, is useful to discover groups and identifying interesting distributions in the underlying data. In most clustering method, the number of clusters, called k, is not clearly known and require user’s assumption, prior knowledge or some practical experience [2]. One of the primary difficulties (if not the primary difficulty) in cluster analysis is determining whether the obtained solution represents the underlying structure in the data or is merely an artifact of the procedure used to obtain that solution. The clustering procedures will always return a k nonempty, non-overlapping, and exhaustive subsets of a data set, regardless of whether that data partitioning provides an appropriate representation of the underlying structure of the data. Figure 1 shows the result of improper choosing of k value. The left-hand panel of Figure 1 indicates a situation where number of centers are not sufficient for data representation. Contrarily, the right-hand panel of Figure 1 illustrates a single cluster that has been erroneously assigned with three different centers; this leads to incorrectly applying a clustering procedure to partition the data when there is only one cluster.

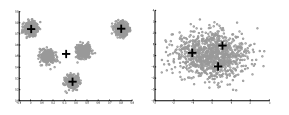


Figure 1: the effect of choosing K value improperly [2]

Center-based clustering algorithms, like k-means and Gaussian expectation maximization, assume each cluster has a unimodal distribution, such as Gaussian [2]; in other word, each subset of data following a unimodal distribution should be considered as one cluster having only one center. Otherwise, multiple centers will less well capture the truth about that unimodal distributed subset of data.

In this project, the G-means algorithm proposed by Hamerly and Elkan [2], is examined. Several data sets were evaluated to examine the accuracy of the G-means in clustering. The k-means algorithm implicitly assumes that the data points in each cluster are spherically distributed around the center. Less restrictively, the G-means algorithm assumes that the data points in each cluster have a multidimensional Gaussian distribution [2]. With this assumption, the problem gets simplified to find a set of Gaussian clusters. To examine if the data points in the same cluster are sampled from Gaussian distribution, Anderson Darling test is used.

# related work

Vanilla K-means clustering has many drawbacks. The major drawback of K-means is that the K in the K-means needs to be learnt. This is a major issue when the data is complicated and has many attributes [3]. The decision for the K in the K-means algorithm is extremely important in determining the quality of the clusters. To solve this problem, one method that can be implemented to learn the K in K-means, is the G-means technique [2].

In the G-means we needed to decide whether the cluster that is divided by the algorithm is a gaussian distribution or not. To do this it is necessary to decide how close the data in the cluster is to an ideal gaussian distribution. This is done using Hypothesis Testing [4] which checks if a cluster is Gaussian or not based on certain Confidence Interval. The value of this confidence interval decides how much precision is needed in the clusters being Gaussian and the how much errors in the hypothesis can be ignored safely.

The test for Gaussian distribution is done using the Anderson Darling test [5], that converts the data into one dimension and checks whether the data is distributed in a Gaussian distribution not based on the probability specified.

The G-means algorithm uses Bisecting K-means clustering in every iteration to divide the clusters further into more clusters [3]. In bisecting K-means every cluster is divided into 2 clusters. Then these clusters are checked for Gaussian distribution. If they follow Gaussian distribution, then they are stored as it is and if they do not follow the Gaussian distribution then they are further divided using bisecting K-means until we get Gaussian distribution, or we reach a minimum cluster size.

Choosing the initial centroids in the k-means is a challenge and the choice of the centroids decides the final clusters. Hence even a very good implementation of k-means algorithms may give us undesirable clusters due to the choice of initial centroids [3]. To overcome this method, the two centroids to be chosen for the bisecting k-means are chosen by finding the biggest variance axis and taking the two farthest points from each other [6]. This ensures that the two clusters formed are well separated and properly bisected.

# Methodology

## Algorithm

The C language implementation detailed out in this report to determine the cluster centroids used during k-means is based on the G-Means algorithm [2].

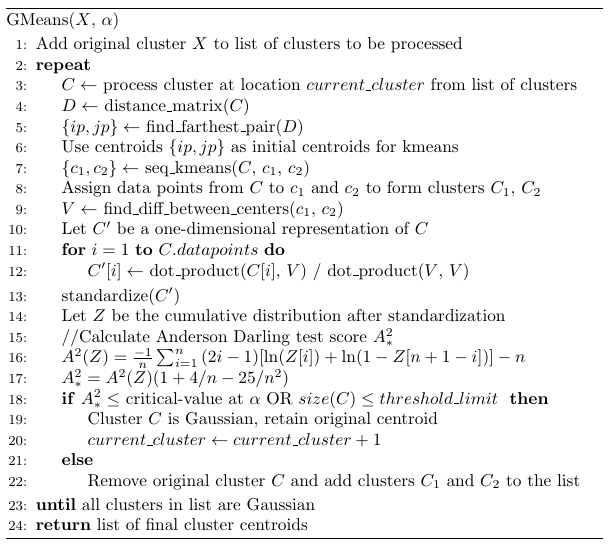


Figure 1. Algorithm based on original paper [2]

## Testing clusters for Gaussian Fit:

As discussed above in algorithm, we need to test whether the cluster is Gaussian distributed. Therefore, Anderson-Darling test is used to detect if data comes from population with Gaussian Distribution. This one-dimensional test has been shown empirically to be the most powerful normality test that is based on the empirical cumulative distribution function (ECDF).

The Anderson-Darling test is defined as:

• H0: The data around the center are sampled from a Gaussian.

• H1: The data around the center are not sampled from a Gaussian

If the cluster passes Hypothesis test, then we believe that one center is sufficient to model the data and there is no need to divide it into two clusters. However, if it fails the hypothesis test, then we reject the current center and accept two new cluster centroids.

## Test Statistics:

Let =F(), where F is the cumulative distribution function with mean 0 and variance 1. Also, xi is the ordered data.

## Preprocessing:

In order to be able to apply the Anderson-Darling test on the data, the multi-dimensional data need to be to be projected onto a single dimension. As suggested by the authors of [2], this first involves connecting the centers of the two potential clusters which will be used in case the original cluster is non-Gaussian. This difference is stored in the vector V. Calculating the dot product between each point of the original cluster C and V and dividing the result by the dot product of V gives the 1-dimensional projection C’. Furthermore, C’ needs to be standardized to contain values with mean = 0 and standard deviation = 1. Finally, the Anderson-Darling test score can be calculated on the normal cumulative distribution of this standardized projection.

# IMPLEMENTATION

## **Specifications**:

|  |  |
| --- | --- |
| System Configuration | Intel i7 6th Gen, 8GB RAM, 2.5 GHz Processor, L3 Cache 4 MB |
| Environment | Eclipse, minGW, Windows 10 |
| Programming Language | C, R |
| Libraries Used | The C Clustering Library [9]  CUDA K-Means Clustering [10] |

## **K\_Means Implementation**:

We used CUDA K-Means Clustering library for k-means.

1.1 **Detection of initial centroids**

Every time k-means is called on the data, we choose two initial centroids as two farthest points on major axis within the subset. We implemented a method ***find\_farthest\_pair*** which makes use of the APIs provided by *The C Clustering Library* in order to achieve this*.*

1.2 **Calling k-means**

seq\_kmeans(data, ncols, nrows, nclusters, centers, threshold, clusterid, &itr)

**Arguments**

data: The data matrix containing multidimensional data points.

ncols: The number of columns in the data matrix.

nrows: The number of rows in the data matrix.

nclusters: The number of clusters k. In our case, it is always 2.

centers: This matrix stores the centroid information.

threshold: Object change membership. In our case it is set to 0.001.

clusterid: The cluster number to which each item belongs.

itr: Number of iterations for k-means.

## **Tranformation to 1- Dimensional data:**

transform\_cluster(transformed\_output\_vector, nrows, v, vector\_product, cluster\_data)

**Arguments**

transformed\_output\_vector: This array stores the transformed 1-Dimensional data.

nrows: The number of rows in the data matrix.

v: v = c1 − c2 is a d-dimensional vector that connects the two centers.

vector\_product: ||v||2

cluster\_data: The data matrix containing multidimensional data points.

## **Data Standardization:**

preprocess (x, n, z)

**Arguments**

x: 1-Dimensional vector returned by transform\_cluster.

n: number of data points.

z: (out) Standardised 1-Dimensional vector with Mean 0 and Variance 1.

## **Find Gaussian Distribution:**

cdf (z, n)

**Arguments**

z: Transformed 1-Dimensional vector with Mean 0 and Variance 1

n: number of data points.

## **Evaluation based on Anderson Darling Test**:

anderson\_darling\_test (z, n)

**Arguments**

z: Sorted 1-Dimensional vector transformed by cdf function.

n: number of data points.

Ouput 🡪

A2∗(Z): score for the given cluster.

# Results

Multiple data sets were used to test the algorithm. Out of these, one of the datasets was a synthetic data set [7] and the other one was data about e-coli bacterium [8].

The goal is to cluster the data without

knowledge of the labels and measure how well the clustering captures the true labels [2]. The synthetic data set has two attributes, 300 data points and 5 true clusters. On the other hand, the e-coli dataset has 7 attributes, 336 data points and 8 true clusters.

The results of the clusters formed are ……….

The algorithm finds 7 clusters for the synthetic data while the true clusters are 5 in the dataset. However, all the clusters predicted are mostly the same grouping as the true clusters.

For the e-coli dataset the algorithm predicts 4 clusters. These are lower than the 8 true clusters. This is because in the true clusters, there are clusters having less than 7 data points. However, the implemented algorithm only keeps clusters that are larger than those and does not divide the data into more clusters if resultant clusters will be smaller than the minimum size.

The execution time for the synthetic data is: ……… . While the execution time for the e-coli data is: …………. .

# Conclusions

The G-means clustering is a much more efficient way to cluster data based on Gaussian distributions in the data, when compared to Vanilla K-means clustering. The important learning objective was to find the K in the K-means so as to get the correct number of clusters based on the data. G-means tends to do a pretty good job in finding the number of clusters in  K-means algorithm. The algorithm stops when the cluster size is too small or when we have Gaussian clusters. This means that the clusters aren’t divided into too small clusters just because the original cluster was not Gaussian in the distribution.

One way of improving the algorithm is preventing the initial clustering itself if the number of data points in the cluster are less than a particular number. In the implementation, the first split always takes place, and then the clusters formed are not split further if the cluster size is too small. However a better implementation can be to prevent the cluster formation itself if either of the two clusters formed is smaller than a specified size.

This implementation assumes that the data available is a set of gaussian clusters from which separate clusters can be extracted. This however may not be true and to overcome this problem, the value of confidence must be set accordingly to overcome the irregularities in the data’s distribution.

# Repository

GitHub: <https://github.ncsu.edu/sjgurav/G-Means>

# Acknowledgement

The authors of the paper want to thank Dr. Ranga Raju Vatsavai (Computer Science Dept, North Carolina State University) for all of his help and support in successfully completing this project.

##### References

1. Tapas Kanungo, Nathan S. Netanyahu, Angela Y. Wu. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7):881–892, 2002.
2. Greg Hamerly, Charles Elkan. Learning the k in k-means. [Advances in Neural Information Processing Systems](http://papers.nips.cc/book/advances-in-neural-information-processing-systems-30-2017), 281-288, 2004.
3. Pan Ning tang, Michael Steinbach, Vipin Kumar. Introduction to Data Mining. Addison –Wesley, Upper Saddle River, NJ, 2005.
4. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, An Introduction to Statistical Learning, Springer Science & Business Media, 2013.
5. George Marsaglia, John C. W. Marsaglia. Evaluating the Anderson-Darling Distribution. Journal of Statistical Software, 9(5), 2004.
6. Ali Ridho Barakbah, Yasushi Kiyoki. A pillar algorithm for K-means optimization by distance maximization for initial centroid designation, [IEEE Symposium on](https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4910529) [Computational Intelligence and Data Mining](https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4910529), 2009.
7. https://github.ncsu.edu/sjgurav/G-Means/blob/master/data/gmean5.csv

[8] UCI Machine Learning library E-Coli data set. (1996, September 1)  
 Retrieved from <https://archive.ics.uci.edu/ml/datasets/ecoli>

[9] The C Clustering Library. (n.d.). Retrieved from   
 <http://bonsai.hgc.jp/~mdehoon/software/cluster>

[10] Giuroiu, S. (n.d.). CUDA K-Means Clustering. Retrieved from   
 [http://serban.org/software/kmeans/](http://serban.org/software/kmeans/%0c)