Hierarchical and Partitioning based Hybridized Blocking Model

by

CHANDRAVYAS ANNAKULA

B.E., Vasavi College of Engineering, 2012

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

College of Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

2017

Approved by:

Major Professor

Dr.William H.Hsu

Copyright

© CHANDRAVYAS ANNAKULA 2017.

Abstract

Improving massive experiments with threshold blocking paper provides us with a sampling blocking algorithm that enables large and complex experiments to run in polynomial time without sacrificing the precision of estimates on covariate dataset. Goal of this project is to run different clustering algorithms on top of clusters formed from above mentioned blocking algorithm and analyze the performance and compatibility of the clustering algorithms.

We first start with applying blocking algorithm on covariate dataset and once the clusters are formed we then apply our clustering algorithm PAM or HAC on the seeds of the clusters. This will help us to generate more similar clusters. We compare our performance and precision of our hybridized clustering techniques with the pure clustering techniques to identify a suitable hybridized blocking model.

Table of Contents

[List of Figures 3](#_Toc479292320)

[List of Tables 4](#_Toc479292321)

[Acknowledgements 5](#_Toc479292322)

[Chapter 1 - Introduction 6](#_Toc479292323)

[Chapter 2 - Literature Survey 7](#_Toc479292324)

[2.1 Hierarchial Agglomerative Clustering Algorithm 7](#_Toc479292325)

[2.1.1 Algorithm 7](#_Toc479292326)

[2.1.2 Pros 7](#_Toc479292327)

[2.1.3 Cons 8](#_Toc479292328)

[2.1.4 Comparision with other other Algorithms 8](#_Toc479292329)

[2.1.5 Complexity 8](#_Toc479292330)

[2.2 PAM Algorithm 9](#_Toc479292331)

[2.2.1 Algorithm 9](#_Toc479292332)

[2.2.2 Pros 9](#_Toc479292333)

[2.2.3 Cons 10](#_Toc479292334)

[2.2.4 Comparison with other clustering models 10](#_Toc479292335)

[2.2.5 Complexity 10](#_Toc479292336)

[Chapter 3 - DataSet 11](#_Toc479292337)

[3.1 Attributes of a dataset 11](#_Toc479292338)

[3.2 Data Filtering 17](#_Toc479292339)

[3.3 Data Validation 17](#_Toc479292340)

[3.4 Feature Selection 18](#_Toc479292341)

[Chapter 4 - Proposed Model 19](#_Toc479292342)

[4.1 HAC with Threshold Blocking Algorithm 20](#_Toc479292343)

[4.2 PAM with Threshold Blocking Algorithm 20](#_Toc479292344)

[4.3 Data Flow Diagram 21](#_Toc479292345)

[4.4 Implementation 21](#_Toc479292346)

[4.5 System Configuration 22](#_Toc479292347)

[Chapter 5 - Experimental Results 23](#_Toc479292348)

[5.1 Cluster Evaluation Measures 23](#_Toc479292349)

[5.1.1 Silhouette Index Value 23](#_Toc479292350)

[5.1.2 Cluster Overlap Measure 24](#_Toc479292351)

[5.1.3 Cluster Distance 25](#_Toc479292352)

[5.1.3.1 Intra-Cluster Distance 25](#_Toc479292353)

[5.1.3.2 Inter-Cluster Distance 26](#_Toc479292354)

[5.2 Experiment1 – Comparing Hybridized HAC with HAC 26](#_Toc479292355)

[5.2.1 Silhouetee Coefficient 26](#_Toc479292356)

[5.2.2 Intra-Cluster Distance 27](#_Toc479292357)

[5.2.3 Inter-Cluster Distance 27](#_Toc479292358)

[5.2.4 Cluster Overlap 27](#_Toc479292359)

[5.2.5 Processing Time 28](#_Toc479292360)

[5.2.6 Memory 28](#_Toc479292361)

[5.3 Experiment2 - Comparing Hybridized PAM with PAM 29](#_Toc479292362)

[5.3.1 Silhouette Coefficient 29](#_Toc479292363)

[5.3.2 Intra-Cluster Distance 29](#_Toc479292364)

[5.3.3 Inter-Cluster Distance 30](#_Toc479292365)

[5.3.4 Inter-Cluster Distance 30](#_Toc479292366)

[5.3.5 Cluster Overlap 30](#_Toc479292367)

[5.3.6 Processing Time 30](#_Toc479292368)

[5.3.7 Memory 31](#_Toc479292369)

[Chapter 6 - Conclusion 32](#_Toc479292370)

[References or Bibliography (choose one) 32](#_Toc479292371)

[Appendix A - Your Appendix Title 34](#_Toc479292372)

[Appendix B - Using appendices, adding more, adding captions 35](#_Toc479292373)

List of Figures

*The List of Figures is created from figure captions in this document. See Chapter 6.* ***1. In your chapters, insert figure captions.*** *You must train them (1, 2, 3, or 1.1, 1.2, 1.3, etc.).****2. Update*** *the list below.* ***Right-click*** *anywhere in the list and select* ***Update Field****.****3. If a window pops up****, select* ***Update entire table*** *and click* ***OK****.*

[Fig 1 Silhouette Comparison for Pure HAC vs Hybridized HAC for various Threshold Values 26](#_Toc479294719)

[Fig 2 Intra Cluster Distance Comparision for Pure HAC vs Hybridized HAC for different threshold values 27](#_Toc479294720)

List of Tables

*The List of Tables is created from table captions in this document. See Chapter 6.* ***1. In your chapters, insert table captions.*** *You must train them (1, 2, 3, or 1.1, 1.2, 1.3, etc.).****2. Update*** *the list below.* ***Right-click*** *anywhere in the list and select* ***Update Field****.****3. If a window pops up****, select* ***Update entire table*** *and click* ***OK****.*

Acknowledgements

The Acknowledgements page is optional. If you include it, retain the Acknowledgements heading and enter your text here.

# Introduction

Improving massive experiments with threshold blocking paper provides us with a sampling blocking algorithm that enables large and complex experiments to run in polynomial time without sacrificing the precision of estimates on covariate dataset. Goal of this project is to run different clustering algorithms on top of clusters formed from above mentioned blocking algorithm and analyze the performance and compatibility of the clustering algorithms.

First the threshold Blocking algorithm is applied on covariate dataset to form clusters. The clusters formed are more similar with less intra cluster distance and more inter cluster distances. Now we apply HAC or PAM clustering algorithm on top of this to form more efficient and similar clusters. Now our main focus would be to compare the performance and compatibility of the clusters formed from HAC in combination with Threshold blocking algorithm against pure HAC Clustering algorithm. Similarly we compare the performance and compatibility of the clusters formed from PAM in combination with Threshold blocking algorithm against pure PAM Clustering algorithm. Thus we are able to analyze the performance and compatibility of the pure clustering algorithms against hybridized clustering algorithms.

# BACKGROUND

Experiments executed with random sample chosen from the data ensure that estimated treatment effects are equal to the true causal effects of interest in expectation. However, the assigned data sample may not be a right fit to test the experiment result or effect. For Example, consider a medical study on the effect that a drug has on life expectancy, it may occur by chance that the control group is older and sicker than the treatment group. In such cases, there is high likelihood to observe inaccurate estimations or results as there are imbalances in covariates. So the studies based on such data contain high variance and the results from the data tend to be biased conditionally on the distribution of covariates.

Unadjusted estimates for even massive experiments are often too variable to enable reliable inferences because the effects of interest may be small and distributional issues result in surprisingly large variances. In the case of massive data, the experiment of interest might be draw fine-grained inferences and targeting the treatments to subgroups. Due to dimensionality curse and random assignment, subgroups of interest used for such experiments might lack sufficient data needed for analysis.

In order to deal with above scenarios, blocking has been the default experimental design to deal with it. With this design, the investigator forms groups of units, or blocks, that are as similar as possible. Treatments are then randomly assigned in fixed proportions within blocks and independently across them. This prevents imbalances in observed covariates, which can increase precision if these covariates are predictive of outcomes. Blocking improves precision in the test result by adjusting for covariates in the design of study rather than from the test result.

Also, existing blocking methods are not sensitive to clustering of data points and are often heuristic. So, the samples generated by these blocking methods doesn’t form a good dataset to the clustering algorithms and thereby leading to erroneous results. Also, the existing algorithms that are proven to be optimal are computationally expensive and especially not feasible for large data sets.

Considering all the above scenarios, the proposed Threshold Blocking algorithm aims to solve all these problems. The algorithm takes an input to threshold value, which is minimum number of points to be contained in each block or group and a distance metric. The algorithm tries to minimize the maximum distance between any two units in the same group. Thus, the algorithm offers flexibility in the block structure and forms blocks resembling natural cluster units which may improve performance.

## 2.1 Algorithm

Given the graph representation of the experimental sample, G = (V, E), and a pre-specified threshold k, the approximate blocking algorithm proceeds as follows:

1. Construct a (k − 1)-nearest neighbor subgraph of G. Denote this graph Gnn = (V, Enn).

2. Find a maximal independent set of vertices, S, in the second power of the (k−1)-nearest neighbor subgraph, G2 nn. Vertices in S are referred to as the block seeds.

3. For each seed i ∈ S, create a block comprised of its closed neighborhood in Gnn, Vi = NGnn[*i*].

4. For each yet unassigned vertex, assign it to any block that contains one of its adjacent vertices in Gnn.

When the algorithm terminates, the collection of blocks, balg = {Vi}i∈S, is a valid threshold blocking of the experimental units that satisfies the optimality bound.

## 2.2 Complexity

The blocking algorithm terminates in T(n) = O(n logk n) using O(kn) space. Currently, available or any of the commonly used blocking algorithms run in polynomial time, but the Threshold blocking algorithm runs in quasilinear time. Moreover, in the case of fixed k and an efficient nearest neighbor subgraph construction algorithm, blocking algorithm runs in O(nlogn) time and O(n) space complexity.

# LITERATURE SURVEY

## 3.1 Hierarchial Agglomerative Clustering Algorithm

This algorithm works by grouping  the data one by one on the basis of the  nearest distance measure of all the pairwise distance between the data point. Again distance between the data point is recalculated but which distance to consider when the groups has been formed? For this there are many available methods. Some of them are:

1) single-nearest distance or single linkage.

2) complete-farthest distance or complete linkage.

3) average-average distance or average linkage.

4) centroid distance.

5) ward's method - sum of squared euclidean distance is minimized.

This way we go on grouping the data until one cluster is formed. Now on the basis of dendogram graph we can calculate how many number of clusters should be actually present.

### 3.1.1 Algorithm

1. Compute the distance matrix between the input data points
2. Let each data point be a cluster
3. Repeat

4. Merge the two closest clusters

5. Update the distance matrix

6. Until only a single cluster remains

### 3.1.2 Pros

1. No apriori information about the number of clusters required.
2. Easy to implement and gives best result in some cases.

### 3.1.3 Cons

1. Algorithm can never undo what was done previously.
2. Time complexity of at least O(*n2log n*) is required, where *‘n’* is the number of data points.
3. Based on the type of distance matrix chosen for merging different algorithms can suffer with one or more of the following:
   1. Sensitivity to noise and outliers
   2. Breaking large clusters
   3. Difficulty handling different sized clusters and convex shapes
4. No objective function is directly minimized
5. Sometimes it is difficult to identify the correct number of clusters by the dendogram.

### 3.1.4 Comparision with other other Algorithms

1. Hierarchical Clustering can give diﬀerent partitionings depending on the level-of-resolution we are looking at where as K-means clustering produces a single partitioning
2. Hierarchical clustering doesn’t need the number of clusters to be speciﬁed
3. Where as K-Means clustering needs the number of clusters to be specified.
4. Hierarchical clustering can be slow (has to make several merge/split decisions) where as K-means clustering is usually more eﬃcient run-time wise

### 3.1.5 Complexity

For a dataset **X** consisting of **n** points  
1. **O(n2)** **space**; it requires storing the distance matrix   
2. **O(n3)** **time** in most of the cases  
3. There are **n** steps and at each step the size **n2** distance matrix must be updated and searched  
4. Complexity can be reduced to **O(n2 log(n) )** time for some approaches by using appropriate data structures

## 3.2 PAM Algorithm

PAM stands for “partition around medoids”. The algorithm is intended to find a sequence of objects called medoids that are centrally located in clusters. The goal of the algorithm is to minimize the average dissimilarity of objects to their closest selected object.

### 3.2.1 Algorithm

It starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resultant clustering. It selects k representative medoid data items arbitrarily. For each pair of non-medoid data item x and selected medoid m, the total swapping cost S is calculated. If S< 0, m is replaced by x. Thereafter each remaining data item is assigned to cluster based on the most similar representative medoid. This process is repeated until there is no change in medoids.

Steps:

1. Use the real data items in the data set to represent the clusters.

2. Select k representative objects as medoids arbitrarily.

3. For each pair of non-medoid item xi and selected medoid mk, calculate the total swapping cost S(ximk). For each pair of xi and mk

If S < 0, mk is replaced by xi

Assign each data item to the cluster with most similar representative item i.e. medoid.

4. Repeat steps 2-3 until there is no change in the medoids.

### 3.2.2 Pros

1. Pam is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean

2. Easy to implement

### 3.2.3 Cons

1. PAM is efficient for small data sets but does not scale well for large data sets

2. Pam works efficiently for small data sets but does not scale well for large data sets. – O(k(n-k)2 ) for each iteration where n is # of data,k is # of clusters

3. Arbitrary Shapes: No! Only Globular Clusters

### 3.2.4 Comparison with other clustering models

The k-medoids method is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean. However, its processing is more costly than the k-mean method.

### 3.2.5 Complexity

# DataSet

The Million Song Dataset is a freely-available collection of audio features and metadata for a million contemporary popular music tracks. The dataset contains only the feature analysis and metadata for one million songs but not the audio provided by [The Echo Nest](http://the.echonest.com/). The size of the entire dataset is around 280GB containing almost one million song records. The features of each record in the dataset consists of the following features.

## 4.1 Attributes of a dataset

|  |
| --- |
| artist\_mbid: db92a151-1ac2-438b-bc43-b82e149ddd50  the musicbrainz.org ID for this artists is db9...  artist\_mbtags: shape = (4,)  this artist received 4 tags on musicbrainz.org  artist\_mbtags\_count: shape = (4,)  raw tag count of the 4 tags this artist received on musicbrainz.org  artist\_name: Rick Astley  artist name  artist\_playmeid: 1338  the ID of that artist on the service playme.com  artist\_terms: shape = (12,)  this artist has 12 terms (tags) from The Echo Nest  artist\_terms\_freq: shape = (12,)  frequency of the 12 terms from The Echo Nest (number between 0 and 1)  artist\_terms\_weight: shape = (12,)  weight of the 12 terms from The Echo Nest (number between 0 and 1)  audio\_md5: bf53f8113508a466cd2d3fda18b06368  hash code of the audio used for the analysis by The Echo Nest  bars\_confidence: shape = (99,)  confidence value (between 0 and 1) associated with each bar by The Echo Nest  bars\_start: shape = (99,)  start time of each bar according to The Echo Nest, this song has 99 bars  beats\_confidence: shape = (397,)  confidence value (between 0 and 1) associated with each beat by The Echo Nest  beats\_start: shape = (397,)  start time of each beat according to The Echo Nest, this song has 397 beats  danceability: 0.0  danceability measure of this song according to The Echo Nest (between 0 and 1, 0 => not analyzed)  duration: 211.69587  duration of the track in seconds  end\_of\_fade\_in: 0.139  time of the end of the fade in, at the beginning of the song, according to The Echo Nest  energy: 0.0  energy measure (not in the signal processing sense) according to The Echo Nest (between 0 and 1, 0 => not analyzed)  key: 1  estimation of the key the song is in by The Echo Nest  key\_confidence: 0.324  confidence of the key estimation  loudness: -7.75  general loudness of the track  mode: 1  estimation of the mode the song is in by The Echo Nest  mode\_confidence: 0.434  confidence of the mode estimation  release: Big Tunes - Back 2 The 80s  album name from which the track was taken, some songs / tracks can come from many albums, we give only one  release\_7digitalid: 786795  the ID of the release (album) on the service 7digital.com  sections\_confidence: shape = (10,)  confidence value (between 0 and 1) associated with each section by The Echo Nest  sections\_start: shape = (10,)  start time of each section according to The Echo Nest, this song has 10 sections  segments\_confidence: shape = (935,)  confidence value (between 0 and 1) associated with each segment by The Echo Nest  segments\_loudness\_max: shape = (935,)  max loudness during each segment  segments\_loudness\_max\_time: shape = (935,)  time of the max loudness during each segment  segments\_loudness\_start: shape = (935,)  loudness at the beginning of each segment  segments\_pitches: shape = (935, 12)  chroma features for each segment (normalized so max is 1.)  segments\_start: shape = (935,)  start time of each segment (~ musical event, or onset) according to The Echo Nest, this song has 935 segments  segments\_timbre: shape = (935, 12)  MFCC-like features for each segment  similar\_artists: shape = (100,)  a list of 100 artists (their Echo Nest ID) similar to Rick Astley according to The Echo Nest  song\_hotttnesss: 0.864248830588  according to The Echo Nest, when downloaded (in December 2010), this song had a 'hotttnesss' of 0.8 (on a scale of 0 and 1)  song\_id: SOCWJDB12A58A776AF  The Echo Nest song ID, note that a song can be associated with many tracks (with very slight audio differences)  start\_of\_fade\_out: 198.536  start time of the fade out, in seconds, at the end of the song, according to The Echo Nest  tatums\_confidence: shape = (794,)  confidence value (between 0 and 1) associated with each tatum by The Echo Nest  tatums\_start: shape = (794,)  start time of each tatum according to The Echo Nest, this song has 794 tatums  tempo: 113.359  tempo in BPM according to The Echo Nest  time\_signature: 4  time signature of the song according to The Echo Nest, i.e. usual number of beats per bar  time\_signature\_confidence: 0.634  confidence of the time signature estimation  title: Never Gonna Give You Up  song title  track\_7digitalid: 8707738  the ID of this song on the service 7digital.com  track\_id: TRAXLZU12903D05F94  The Echo Nest ID of this particular track on which the analysis was done  year: 1987  year when this song was released, according to musicbrainz.org |

Since the project aims to identify similar songs to group them into genres, only few fields among all the above fields are sufficient for the task. Loudness, Tempo, Time\_Signature, Duration and Key are the fields that will be used in this project. So, the million records consisting only of these fields is used in the Experiment.

## 4.2 Data Filtering

The data is distributed using hdf5 files which are converted to .csv extension files using python wrapper. The created .csv files are further filtered to retrieve only the required parameters for the analysis of given problem. Thus, from the .csv files the fields loudness, tempo, time\_signature, duration and key are filtered to form the dataset used in the prediction task.

## 4.3 Data Validation

In the holdout method, we randomly assign data points to two sets d0 and d1, usually called the training set and the test set, respectively. The size of each of the sets is arbitrary although typically the test set is smaller than the training set. We then train on d0 and test on d1. In typical cross-validation, multiple runs are aggregated together; in contrast, the holdout method, in isolation, involves a single run. While the holdout method can be framed as "the simplest kind of cross-validation", many sources instead classify holdout as a type of simple validation, rather than a simple or degenerate form of cross-validation. We start from 90% training data and 10% test data. The training data size percentage is decreased by 10% and test data set size is increased by 10% in each iteration of hold-out cross-validation. This is carried out until we reach 10% training data and 90% test data.

Hold out cross-validation is done against the threshold values suitable for both HAC and PAM algorithm to run on entire 1 Million data set. The below process is illustrated for 1 run of cross-validation among the 10 folds. This process remains same for the rest of the folds but the size of training and test data set changes with each fold.

1. In the project, training data and test data are formed out of the samples of the dataset.
2. The hybridized algorithm model is trained on the training data and thus, formed model is used to predict the cluster number for the testing data. The process of prediction will not affect in changing the cluster center formed out of the model.
3. The inter-cluster and intra-cluster average distances for the clusters are used as measures to validate the system. These measure are used to validate the model.
4. These measures are calculated for the clusters that are formed from training data is validated against the clusters that are formed after merging the test data with training data model clusters.
5. The measures are to be similar in order to avoid overfitting of the model.

## 4.4 Feature Selection

The original Million Song dataset donot contain any labels or genre information. The goal of clustering task in the project is to predict the genre label of each song. From background study related to music data it is identified that there are 13 features in million song dataset that is helpful for us to classify the given song into a genre group but there is a possibility that some features may be redundant so as part of feature selection highly correlated features in the dataset is identified and removed. At the end only 5 features are identified that provide us with unique information to classify the clusters into genre groups.

# Proposed Model

In our model we will be using the threshold blocking algorithm as out preprocessing step of actual clustering algorithm to see how efficiency and accuracy of clustering algorithm is strengthened by using threshold blocking algorithm. In this paper we will be frequently using the term Hybridized blocking model which means that Threshold blocking algorithm is combined with either PAM or HAC to perform clustering with Threshold Blocking algorithm as the preprocessing step. Performance metrics are evaluated by comparing the hybridized blocking algorithm with pure clustering algorithms which in our case is PAM and HAC. We first pass the same dataset with same size to Hybridized clustering algorithm as well as the pure clustering algorithms and once we get the cluster assignment from the both the approaches we then evaluate the approach against parameters like like inter-cluster distance, intra-cluster distance, Silhoutte Coefficient and similarity between the cluster outputs.

In our project we will be carrying two main experiments:

* Threshold Blocking Algorithm with HAC Vs HAC.
* Threshold Blocking Algorithm with PAM Vs PAM.

Since Threshold blocking expects the dataset to be covariate we are considering Million Song Dataset to evaluate the performance. Million song dataset has as many as 48 features but we will be considering only 5 features which would help us to generate clusters of similar songs.

The clusters formed by running the clustering algorithms represents different genres. As the dataset contains only 13 genres, we run the clustering algorithms to divide the data into 13 clusters.

Initially, Million Song dataset is given to threshold blocking algorithm to form clusters such that each cluster contains minimum number of elements specified by threshold value. These samples are closely connected points in multi-dimensional space. The threshold value ensures that data is divided into samples, where each sample consists of points with high similarity measure between any two points in the sample. The centroid calculated for the sample represents the characteristics of sample as a whole. The centroids calculated from each of these clusters is given to both HAC Algorithm and PAM Algorithm. So, the project consists of two parts. Firstly, we analyze the performance and validate the results of hybridized algorithm consisting of Threshold Blocking Algorithm and HAC Algorithm. Secondly, the same steps are repeated against ThresholdBlocking Algorithm and PAM Algorithm.

## 5.1 HAC with Threshold Blocking Algorithm

The Million Song Data set is initially clustered using a random k value by Threshold Blocking Algorithm. The centroids of the above clusters formed out of this algorithm is given as input to HAC Algorithm. The HAC algorithm is made to divide these centroids into 13 clusters where each cluster representing the genre. The centroid of the sample and the points corresponding to the sample are clustered into the same cluster consisting of the centroid of the sample. Since, the centroid of the sample represents it as a whole, the points of the sample as well can be clustered into the same cluster as the centroid. Thus, all the records of the data set are divided into 13 clusters.

On the other hand, the entire data set is given to HAC for cluster analysis. The clusters thus formed using HAC are compared against the clusters formed by above hybridized algorithm to check how many points overlap and how many points donot overlap. Also, with various values of k, the change in intra-cluster and inter-cluster distances, time, memory and other such cluster evaluation factors are used to depict the performance of hybrid algorithm.

## 5.2 PAM with Threshold Blocking Algorithm

In the PAM, the first step of sampling based on the k value remains same as above. The Million Song Dataset in divided into samples or clusters consisting of minimum k points in each sample. The centroids of these samples are passed to PAM for analysis.

Cluster evaluation metrics like intra-cluster and inter-cluster distances, time, memory are calculated for the generated clusters. These metrics are calculated for every instance of k value that is passed to algorithm. A range of k values are chosen to be given as input to the threshold blocking algorithm like in PAM to check the performance variance over various values of k.

PAM is also run on the dataset without any processing step of threshold blocking algorithm. The clusters thus generated are used to compare the similarity with the clusters generated by PAM and threshold blocking algorithm. The metrics of generated clusters are also computed which are compared with the hybridized algorithm for every instance of k.

## 5.3 Data Flow Diagram



## 5.4 Implementation

* Million Song Dataset is obtained from the following source. <https://labrosa.ee.columbia.edu/millionsong/pages/getting-dataset>
* The files provided by the dataset are in .h5 format which are converted using python wrapper code into .csv files.
* Data is pre-processed by removing the covariate variables and features obtained by running Feature Selection Algorithm are retrieved either from the .csv files or during the conversion from .h5 to .csv files. This forms the dataset for the project.
* Dataset consisting of Million records is used for running the hybridized K-Means, hybridized DBSCAN, DBSCAN and K-Means algorithm.
* Due to computation limits, entire data set for clustering is given to hybridized K-Means and K-Means Algorithm while a random subset is chosen for clustering to DBSCAN and hybridized DBSCAN. It is cross-validated against the K-Means model before running the experiment.
* R wrapper of the Threshold Blocking Algorithm provides an implementation of the algorithm. This library is used to initially run on the dataset chosen for the experiment i.e., 1 million record dataset for K-Means and 30,000 record dataset for DBSCAN. Given a threshold value k, the algorithm divides the dataset into blocks which consist of minimum k points.
* For each of these clusters formed out of Threshold Blocking Algorithm, centroids are calculated such that it represents the block as a whole. These centroids are written to another file which is given as input to K-Means or DBSCAN algorithm for clustering.
* In Experiment1, the output from Threshold Blocking algorithm in given to K-Means algorithm for clustering. On the other part, the data set which is given to Threshold Blocking Algorithm is given to K-Means to compute the clusters for the data set. The output from hybridized K-Means algorithm is compared against the output of K-Means algorithm to evaluate the performance of the algorithm.
* In Experiment 2, the output from Threshold Blocking algorithm in given to DBSCAN algorithm for clustering. On the other part, the data set which is given to Threshold Blocking Algorithm is given to DBSCAN to compute the clusters for the data set. The output from hybridized DBSCAN algorithm is compared against the output of DBSCAN algorithm to evaluate the performance of the algorithm. Before running the DBSCAN algorithm, based on the knndistPlot parameter values are estimated.
* The above two experiments are carried out for various threshold values given to Threshold Blocking algorithm and evaluated against various metrics.

## 5.5 System Configuration

**Operating System:-** Windows 64-bit Operating System

**Programming Language:-** R

**RAM:-** 32 GB Memory

**Processor:-** i7-6700K CPU@4.00GHz

# Experimental Results

The two experiments

1) Comparing Threshold Blocking Algorithm with HAC Vs HAC and

2) Comparing Threshold Blocking Algorithm with PAM Vs PAM.

Threshold Blocking Algorithm is run for various values of K say 5,10,15,20,50,60,70,80,90,100,150,200,250. Due to computational limits, only some among the K values are run. This is because, if the centroid data points data set is too large for the clustering Algorithm like PAM and HAC to run, it takes infinite amount of time to figure out the clusters. So a random subset of data is taken to measure the below Cluster Evaluation measures as well as the performance metrics of the algorithm.

## 6.1 Cluster Evaluation Measures

For an Unsupervised Clustering approach, the evaluation measures are not based on ground truth or on comparison with true label. It is based on the separation of data into clusters. Various indexes and metrics are present to evaluate the performance of the algorithm based on how efficiently an algorithm can separate the data into Clusters. Silhouette Coefficient, Calinski-Harabaz Index are some of the examples. Silhouette Coefficient is considered as the standard index among them.

### 6.1.1 Silhouette Index Value

The below table summarized the range of values taken by the index value measure when run on the clustering output and the interpretation of value related to the performance of the algorithm.

|  |  |
| --- | --- |
| Range of SC | Interpretation |
| 0.71-1.0 | A strong structure has been found |
| 0.51-0.70 | A reasonable structure has been found |
| 0.26-0.50 | The structure is weak and could be artificial |
| < 0.25 | No substantial structure has been found |

### 6.1.2 Cluster Overlap Measure

Cluster overlap measure determines how many clusters overlap between two clustering algorithm outputs. The overlap of two clusters i.e., cluster1 output from clustering algorithm1 and cluster1 output from clustering algorithm2 is calculated by the number of points in cluster1 of algorithm1 that are also present in the cluster1 of clustering algorithm2.

Given two clustering algorithm outputs, in an unsupervised approach the numbers from both the algorithms donot necessarily talk about the same cluster. For Example, cluster 1 from the output of Algorithm1 can relate to the cluster 3 of Algorithm 2. In such a case comparing the number of points in Cluster 1 of Algorithm1 present in Cluster1 of Algorithm2 is not correct and also leads to erroneous results. Also, with each iteration of the algorithm the clusters numbers are randomly assigned to the data set.

To avoid this, the following procedure is carried out in determining the cluster overlap measure:-

* A matrix is constructed out of clustering algorithm outputs where the row on the top corresponds to cluster number of Algorithm1 and column on the left contains the cluster number of Algorithm2.
* The matrix contains values of how many elements match between the cluster outputs from both the algorithms.
* At the intersection of row *i* and column *j*, the value of the cell *ij* gives the information about how many elements of clusteri matches with elements of clusterj.
* For each row *i* , the maximum value among the intersection of row *i* and various values of column *j* is identified. Cluster *i* and Cluster *j* are assumed to be representing the same cluster.
* The same process is carried out for rest of the rows as well and the cluster number represented by the row is matched with some column with which it shares maximum number of elements.
* At the end of nth row, the column value assignments of all the rows have to be distinct. That is each cluster number represented by the column is assigned to one of the cluster number represented by row.

In some cases, at the end of nth row it is possible for one column cluster number to be assigned to more than one row cluster number. It is possible in this case, that a column cluster number is not assigned to any cluster number represented by the row. In such case, use backtracking to assign the column cluster number to row by minimizing the error value. Continue this process until all column cluster numbers are assigned to row cluster numbers and the error is minimized while maximizing the throughput.

Since, the relation between the cluster output labels given by both the algorithms is determined they are compared like in the case of Supervised algorithm. One of the cluster outputs is replaced with the mappings obtained from the above algorithm so as to have a baseline to compare both the algorithms. One output of the cluster acts as the ground truth while the other output values of clustering algorithm are evaluated against it. Hence, we obtain the cluster overlap measure between both the clusters.

### 6.1.3 Cluster Distance

Algorithms that produce clusters with low intra-cluster distances have high intra-cluster similarity and high inter-cluster distances have low inter-cluster similarity. Such a clustering algorithm that produces a collection of clusters having low intra-cluster distance and high inter-cluster distance is considered as the best algorithm based on this criterion.

#### 6.1.3.1 Intra-Cluster Distance

The intra-cluster distance *d* '(*k*) is measured as the maximal distance between any pair of elements in cluster *k.*

#### 5.1.3.2 Inter-Cluster Distance

The inter-cluster distance *d*(*i*,*j*) between two clusters may be any number of distance measures, such as the distance between the centroids of the clusters.

## 6.2 Experiment1 – Comparing Hybridized HAC with HAC

A random subset from 1 Million Song data set is given as input to both Hybridized HAC algorithm and HAC Algorithm. The K- value for HAC denotes the number of clusters to be formed which is 13. Since, the data set contains collection of songs from 13 unique genres as per the source, this value is chosen as K input to HAC. The output is 13 different Clusters where each Cluster contains similar songs. The below performance metrics of the algorithm for computation and cluster evaluation are collected over various values of “k”. The “k” value represents the threshold value given to Threshold Blocking Algorithm. When k=0, it implies the data set is run on HAC itself. The chosen Threshold values for the experiment are 10,15,20,25,50,60,70,80,90,100,150,200,250.

### 6.2.1 Silhouette Coefficient

Here K0 represents the silhouette coefficient for pure HAC on random subset of million song dataset. And other K values represents the Silhouette coefficient for Hybridized HAC. For K0 silhouette is 0.23 but as K value increases the silhouette values increased to 0.259 at K20 which represents that the clusters formed represents near to the actual clusters. Later the value starts decreasing as K rises again but the decrease is not lesser than the K0 silhouette value . But as the K raises again the silhouette value again increased and reached to value of .257 at k80 and 0.247 at K250. From this it is clear that hybridized HAC is working better than Pure HAC for different K values and near real clusters are formed by Hybridized HAC when compared to real HAC.

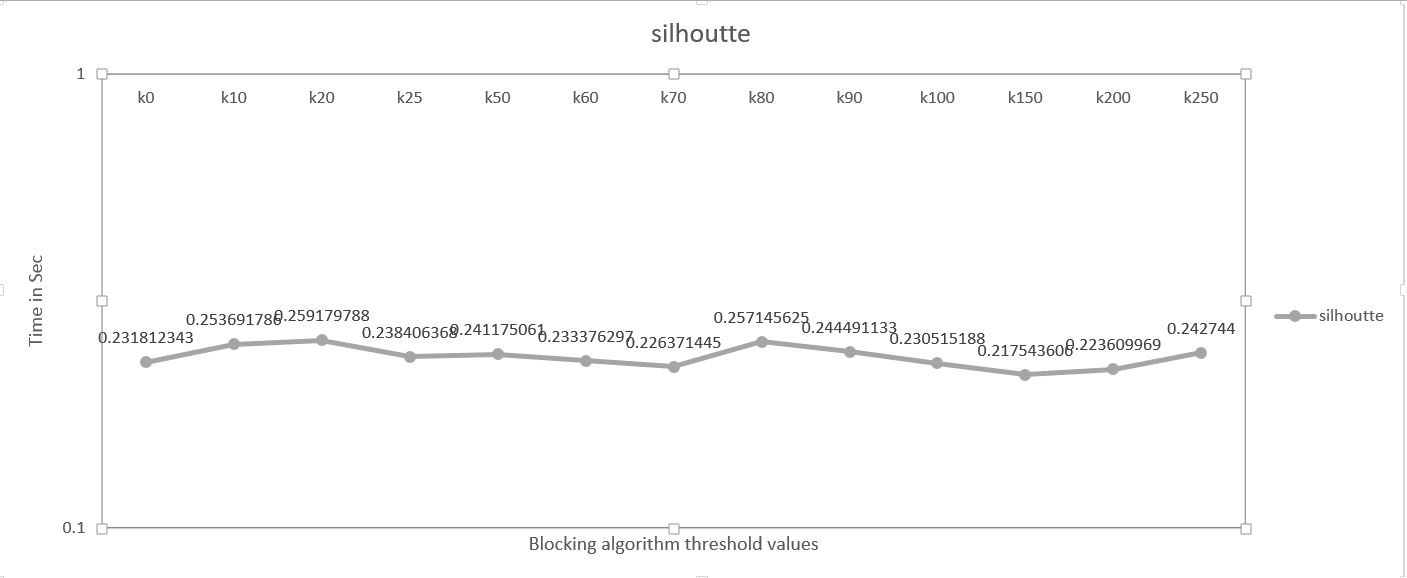
****

Fig 1 Silhouette Comparison for Pure HAC vs Hybridized HAC for various Threshold Values

### 6.2.2 Intra-Cluster Distance

Average intra cluster distance is taken and plotted against various values of k. For k=0 i.e., on original HAC algorithm, the value is around 70. This average distance increases initially until threshold value k =10 which is being run on hybridized K-Means algorithm, but later decreases and remains the same having a value around 40. This indicates there is good high intra-cluster similarity in the clusters obtained from the hybridized algorithm when compared to original HAC algorithm.

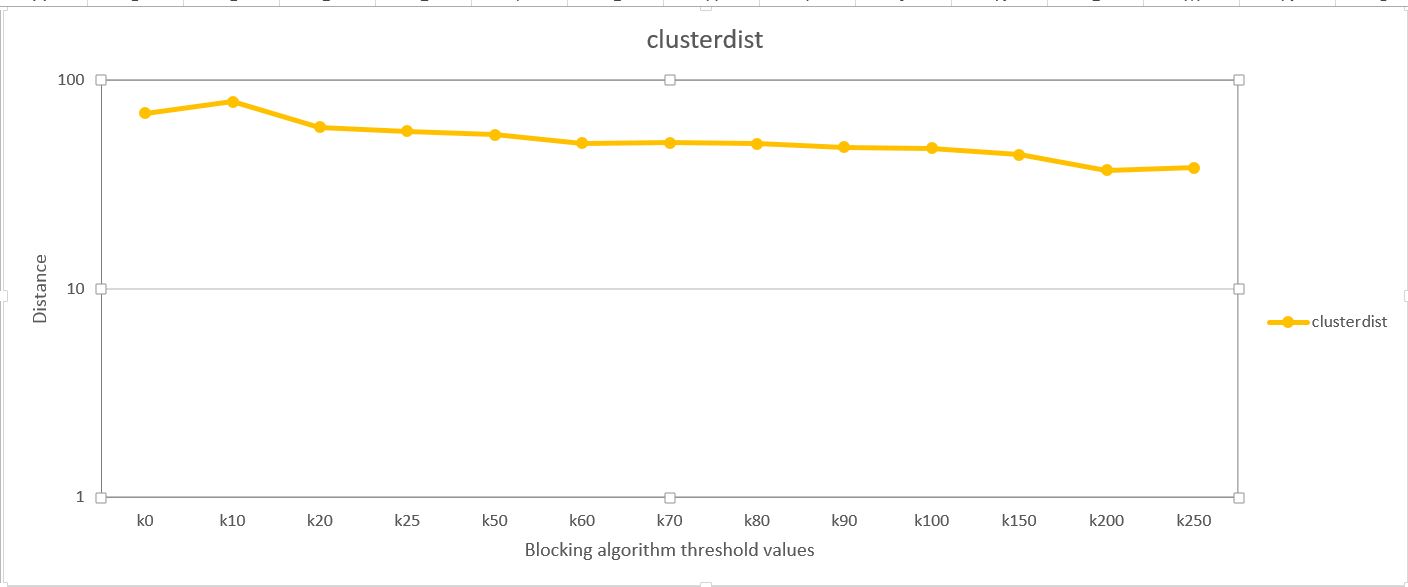
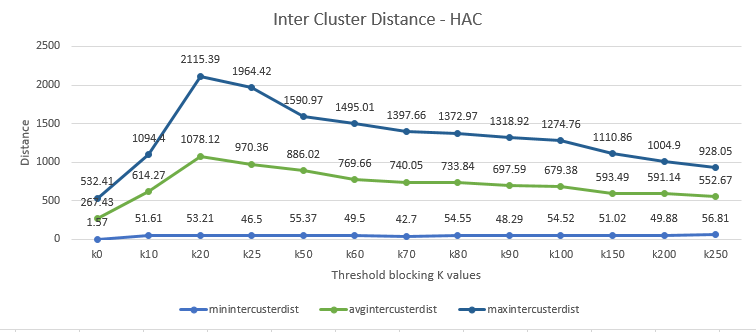
****

Fig 2 Intra Cluster Distance Comparision for Pure HAC vs Hybridized HAC for different threshold values

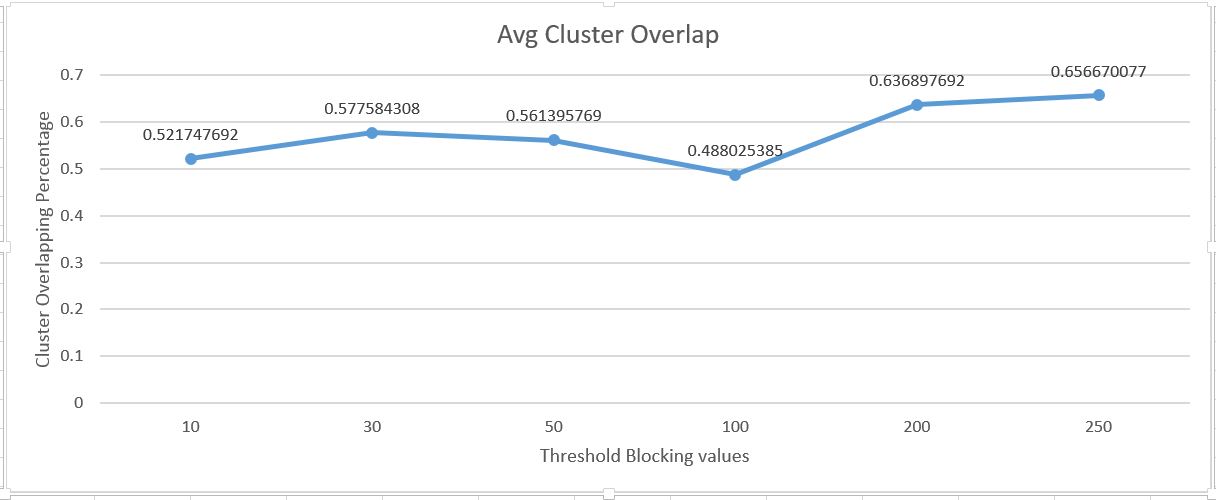
### 6.2.3 Inter-Cluster Distance



From the above observation it is very clear that there is a sharp rise in inter cluster distance from K0 to K10 and the raise is maintained till K20 and after which there is a decline in inter cluster distance but however the value of inter cluster distance in case of hybridized algorithm is very better than Pure HAC. From the overall observation we can say that better clusters are formed on using hybridized algorithm.

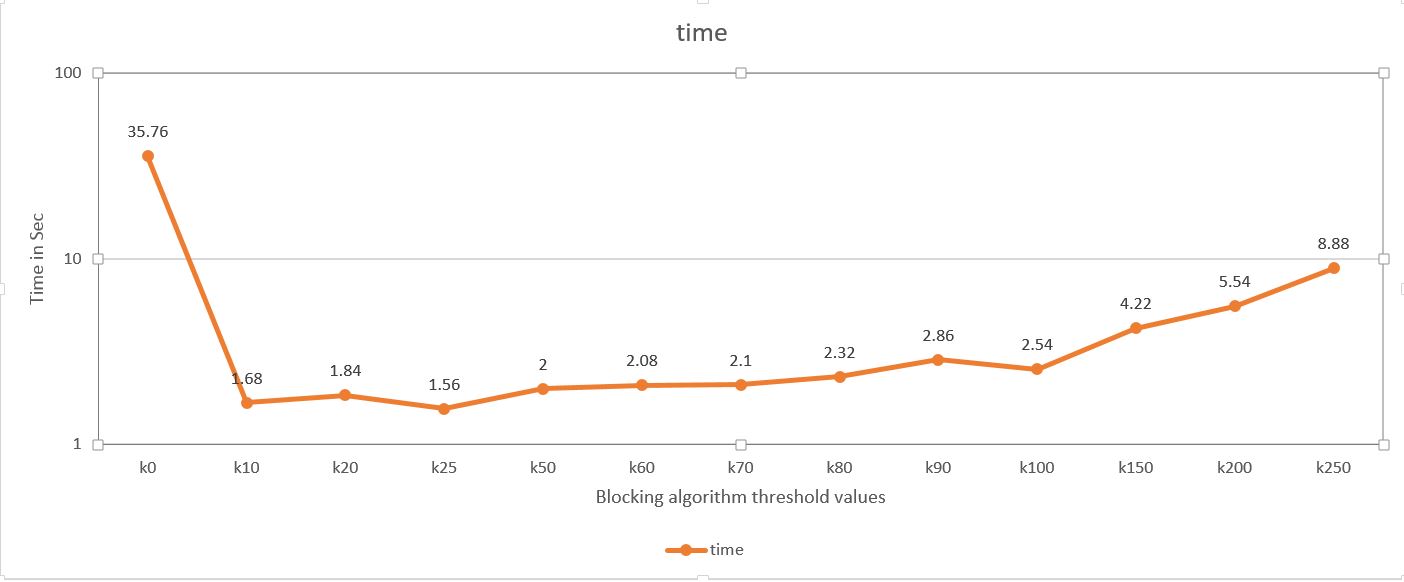
### 6.2.4 Cluster Overlap

On an average it is observed that more similar clusters are formed with higher values of K.



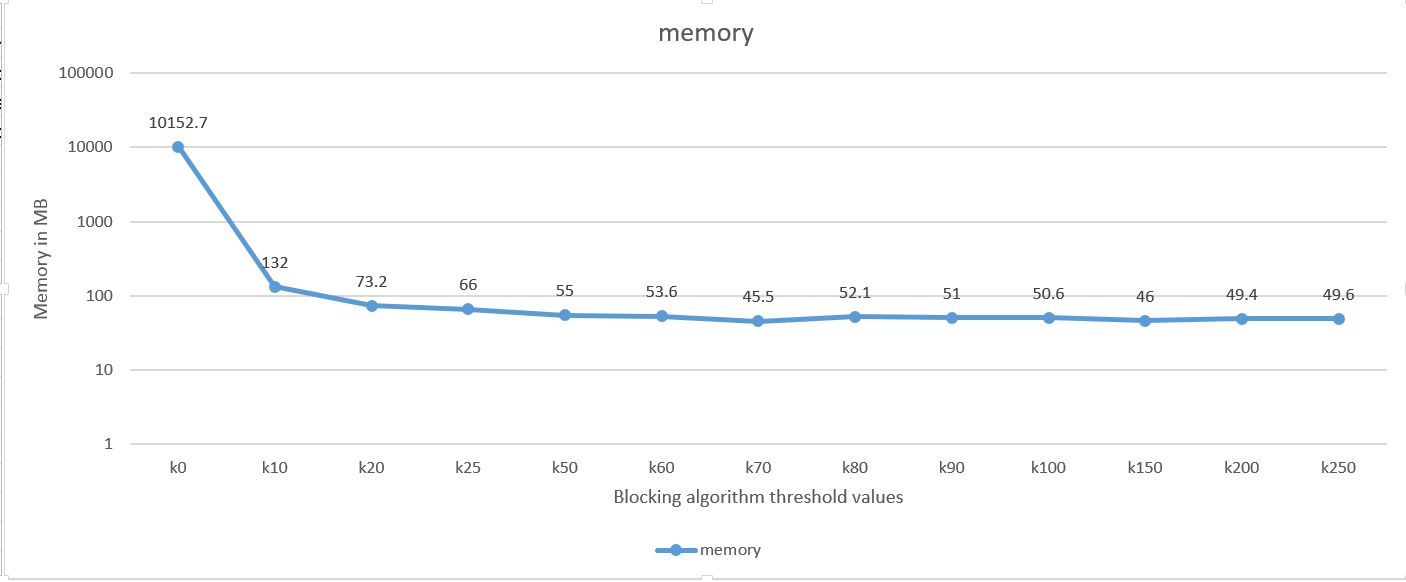
### 6.2.5 Processing Time

There is a drastic decrease in processing time for Processing given data using Pure HAC (K0) compared to hybridized HAC (Other K Values). The decrease is almost 77% which is a good result this this decrease in processing time is not sacrificing the accuracy of the results in turn increasing the accuracy of the results as observed in Cluster Overlap and Silhouette Coefficient results.

****

### 6.2.6 Memory

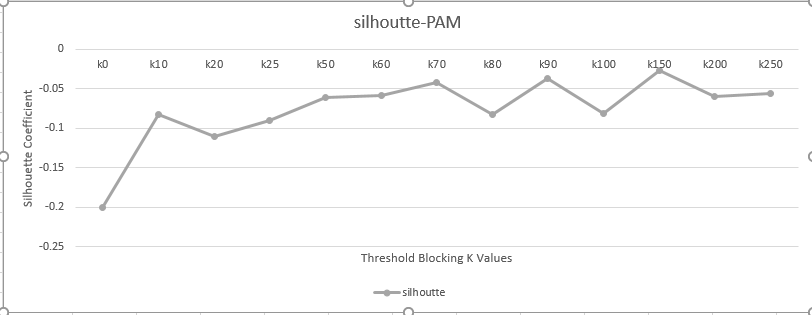
The memory required to run the hybridized HAC algorithm is comparatively low when compared to the memory required to run the Original HAC algorithm on the dataset. This could be because, the size of dataset reduces after the formation of blocks by Threshold Blocking Algorithm and only the centroids are given to HAC after that step. The Threshold Blocking algorithm do not seem to occupy much memory to form the blocks.

****

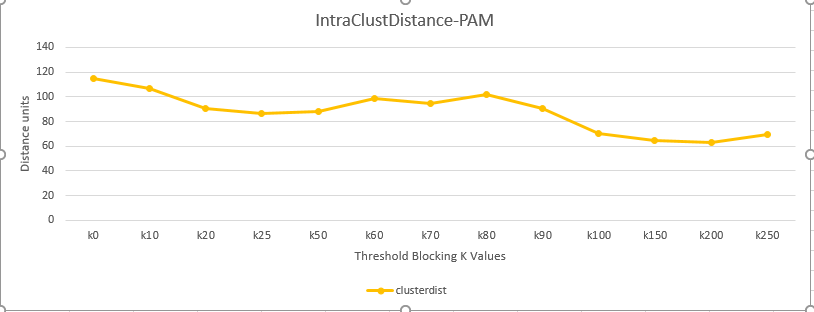
## 6.3 Experiment2 - Comparing Hybridized PAM with PAM

A random subset from 1 Million Song data set is given as input to both Hybridized PAM and PAM Algorithm. The K- value for PAM denotes the number of clusters to be formed which is 13. Since, the data set contains collection of songs from 13 unique genres as per the source, this value is chosen as K input to PAM. The output is 13 different Clusters where each Cluster contains similar songs. The below performance metrics of the algorithm for computation and cluster evaluation are collected over various values of “k”. The “k” value represents the threshold value given to Threshold Blocking Algorithm. When k=0, it implies the data set is run on PAM itself. The chosen Threshold values for the experiment are 15,20,25,50,60,70,80,90,100,150,200,250.

### 6.3.1 Silhouette Coefficient



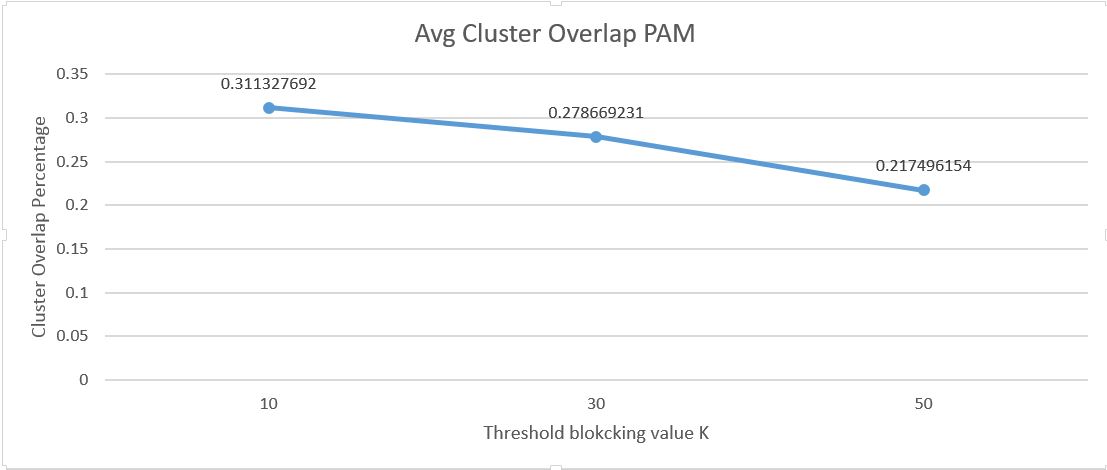
### 6.3.2 Intra-Cluster Distance



### 6.3.3 Inter-Cluster Distance

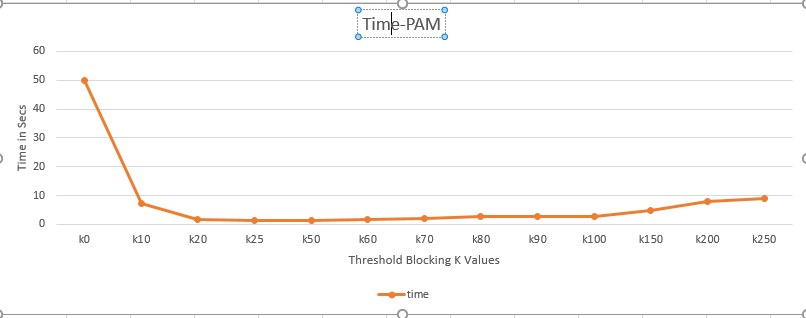
### 6.3.4 Cluster Overlap

As K value increases the Cluster Overlap between clusters formed in Pure PAM vs clusters formed using Hybridized PAM kept decreasing.

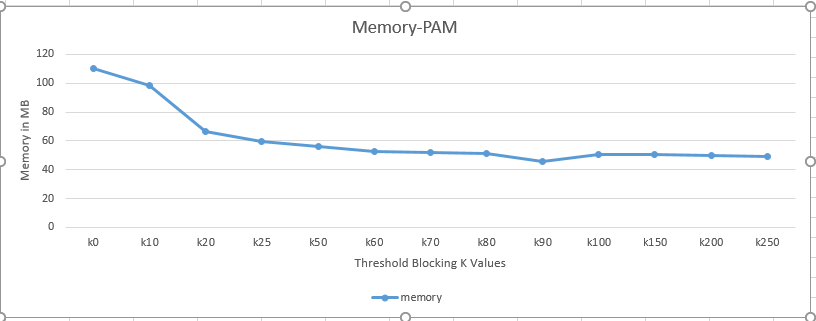


### 6.3.5 Processing Time

The processing time of the hybridized PAM algorithm remains significantly low when compared with PAM for K values below 100. But however as K values raises greater than 100 the processing time kept increasing and processing time shows a greater difference when compared lower values of K.



### 6.3.6 Memory



# Conclusion

From the above experiments hybridized HAC algorithm is proved to perform better than Original HAC. PAM or Hybridized PAM is not suitable for the million song dataset as it gives negative silhouette coefficient which means that the clusters formed using PAM is no way near to real clusters so it is identified that PAM is not the right algorithm for million Song dataset. In terms of memory and processing time the, the hybridized algorithms show a significant drop which indicates the capability of threshold blocking to be extensible to perform clustering on Large datasets.

HAC is initially not feasible to execute on Million song dataset while also taking large amount of time to run on the subsets of the data. When combined with threshold blocking algorithm, there is drop in the time and memory taken for execution with overall increase in accuracy of clustering results. The same applies to K-Means and DBSCAN. This is tested for 1 million data but the experiments are limited by computing demand of algorithms which calculates metrics on the output of clusters. The distance metric that is calculated for 1 million data requires a huge RAM around 2500 GB. Ideally, such an amount of computing memory is not required to generate clusters or for calculating metrics on 1 million data set. To avoid this problem, the computation of metrics, cross-validation have to be calculated using map-reduce algorithms executed on Bigdata technologies like Hadoop, Spark and soon. The performance of threshold blocking algorithm in association with clustering algorithms is tested over various values of K but the performance of threshold blocking algorithm for different sizes of data sets is yet to be explored. The limit of dataset size that threshold blocking algorithm can efficiently handle needs to be calculated.

References

1. Introduction to Data Mining by Pang-Ning Tan, Michael Steinbach, Vipin Kumar.
2. <http://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf>
3. <http://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation>
4. [https://en.wikipedia.org/wiki/Hierarchical\_clustering](https://en.wikipedia.org/wiki/Hierarchical_clustering%20)
5. <http://sekhon.berkeley.edu/papers/algorithmthreshold.pdf>
6. <https://en.wikipedia.org/wiki/K-medoids>
7. Least squares quantization in PCM <http://ieeexplore.ieee.org/document/1056489/>
8. Berkeley Symposium on Mathematical Statistics and Probability <http://projecteuclid.org/euclid.bsmsp/1200512992>
9. Comparisons between Data Clustering Algorithms <http://iajit.org/PDF/vol.5,no.3/15-191.pdf>
10. <https://en.wikipedia.org/wiki/Cluster_analysis>
11. <https://en.wikipedia.org/wiki/Silhouette_(clustering)>
12. "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". Computational and Applied Mathematics
13. <http://nlp.uned.es/docs/amigo2007a.pdf>
14. <https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set>
15. <https://pdfs.semanticscholar.org/d92f/76db8442530139359b803df030a40cba4aad.pdf>
16. <https://en.wikipedia.org/wiki/Cross-validation_(statistics)>

###### Your Appendix Title