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

Abstract

(Higgins, Savje, & Sekhon, 2016) 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 a covariate dataset. The goal of this project is to run the 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 the blocking algorithm on a covariate dataset and once the clusters are formed, we then apply our clustering algorithm HAC (Hierarchical Agglomerative Clustering) or PAM (Partitioning Around Medoids) 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

Acknowledgements

I would like to express my deepest gratitude to my major professor Dr. William H. Hsu for his constant input, comments, and feedback while working on this project and for his constant support during my Master's at Kansas State University. I would like to thank Dr. Mike Higgins, Assistant Professor in the department of Statistics for allowing me to work on his algorithm and for taking time to mentor me throughout the project without which this project might not have happened. I would also like to thank Dr. Mitch Neilsen and Dr. Torben Amtoft for serving on my M.S. committee. Finally, I thank my family and friends for their love and support.

Chapter 1 - Introduction

(Higgins, Savje, & Sekhon, 2016) 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 a covariate dataset. The goal of this project is to run the different clustering algorithms on top of clusters formed from above mentioned blocking algorithm and analyze the performance and compatibility of the clustering algorithms.

The project goal is to evaluate the performance of threshold blocking algorithm as well as the variation in performances of clustering algorithm when combined with threshold blocking algorithm. The data is chosen in order to achieve this goal is sufficiently large data set which is Million Song Dataset. Clusters formed out of the threshold blocking algorithm represent similar units. Clustering on such similar units heuristically is less overloaded when compared to clustering on the complete data set. The threshold blocking algorithm gives us clusters of similar units when Million Song Dataset is given as input. The centroid representing the similar units of these clusters represent the similar units as a whole and thereby eliminating the step of processing extra points. This is because the centroid of the similar unit or block formed out of the algorithm are supposed to portray the characteristics of the points in the block. In the next step, these centroids formed out of threshold blocking algorithm are given to clustering algorithms.

HAC (Hierarchical Agglomerative Clustering) and PAM (Partitioning Around Medoids) are chosen for this project. The subset of Million Song Dataset is given as input to these algorithms to generate clusters due to performance limitations. On the other hand, hybridized algorithms are formed out of threshold blocking algorithm and either of the HAC or PAM. In the hybridized algorithm, the first step requires generating the clusters formed out of threshold blocking algorithm and in the next step the centroid from the clusters thus formed is given as the data source to HAC or PAM. Once the centroids are clustered using the existing clustering algorithms, the cluster represented by the centroid of the similar unit is the cluster for all the data points of the cluster. The cluster outputs from each of the hybridized versions is evaluated against its original version of execution that is hybridized PAM is evaluated against PAM on the same subset of the data and

hybridized HAC is evaluated against HAC on the same subset of data. This helps us in evaluating the performance of hybridized algorithm over the original clustering algorithm.

The second goal of the project is also to gauge the performance limitations of threshold blocking algorithm over various values of k i.e., threshold value. The experiment of hybridized algorithm over the original clustering algorithm is set up and hybridized algorithm is executed for each value of k among the chosen threshold values. The efficiency in forming these clusters and computational variances observed for various values of k are noted down from the experiments in order to evaluate the performance of the algorithm. Since the clustering being done in the project is unsupervised learning, the cluster evaluation measures like inter-cluster distances, intra-cluster distances, silhouette index, cluster overlap measure and other computational parameters like memory, processing time are also chosen to compare the performance of clustering outputs.

The clustering measures of hybridized algorithms are almost around the same value when compared with original algorithm, but the computational resource requirements like processing time, memory are drastically reduced and thereby, providing an improvement over the original algorithm. For algorithms like PAM which are not scalable for large data set, forming a hybridized algorithm with threshold blocking algorithm improves the performance of the algorithm significantly. With increasing values in K, the threshold blocking algorithm performance or computational resource requirement is increased which can be attributed to the sparse structure of the obtained random subset of the data.

The document outlines the algorithms in general and discusses their limitations. Following that it is explained how the data is prepared to be passed to the algorithm and how the experiment is setup for performance evaluation of the algorithm. Lastly, the comparison of the cluster outputs of the algorithms is done and possible inferences are drawn. The project in the end also discusses about the possible limitations encountered by the project and the alternatives available to bypass those limitations in order to increase the scope of the experiment and to evaluate it.

Chapter 2 - 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. Therefore, 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 the curse of dimensionality and random assignment, subgroups of interest used for such experiments might lack sufficient data needed for analysis.

Blocking has become the default experimental design of choice for dealing with the above scenarios. 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.

In addition, existing blocking methods are not sensitive to clustering of data points and are often heuristic. Therefore, the samples generated by these blocking methods does not form a good dataset to the clustering algorithms and thereby leading to erroneous results. In addition, 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. One more advantage of threshold blocking algorithm when compared to fixed size blocking is that in the case of fixed size we might not respect natural clustering of units and one is sometimes forced to assign similar units to different blocks just to satisfy the cardinality condition where as in the threshold blocking we can specify the number of units a cluster should contain based on the type of data thus respecting natural clustering of units.

2.1 NP-Hardness of Threshold Blocking Problem

We consider the blocking problem where one wants to minimize the greatest within-block Dissimilarity, as measured by an arbitrary distance metric, subject to a minimum required block size. Solving this is an NP-Hard Problem. Let us see why this is an NP Hard Problem.

Let k denote a threshold for the minimum block size. Consider the complete graph G = (V, E) describing an experimental sample, where V denotes the set of n vertices (the experimental units) and E denotes the set of edges connecting all pairs of vertices. For each $ij \in E$ there is an associated cost, c_{ij} , indicating the dissimilarity between i and j; lower costs mean that units are more Similar. We require that these costs satisfy the triangle inequality:

$$\forall ij, jl, il \in E\ , c_{ij} + c_{jl} \geq c_{il} \qquad (1)$$

This ensures that the direct route between two vertices is no longer than a detour through a third vertex. All distance metrics fulfill this criterion by definition.

Definition 1: A threshold blocking with threshold k is a partition $b = \{V_1 \ldots V_m\}$ of V where each block satisfies the size threshold:

$$\forall\ V_x \in b, |V_x| \geq k \qquad (2)$$

Definition 2: The subgraph generated by a blocking $b = \{V_1 \dots V_m\}$, denoted G(b) = (V,E(b)), is the union of subgraphs of G induced by the components of b; that is, an edge $ij \in E(b)$ only if i and j are in the same block:

$$E(b) \equiv \{ij \in E: \exists V_x \in b, i, j \in V_x\} \quad (3)$$

Let Bk denote the set of all possible threshold blockings of G with a threshold of k. The bottleneck threshold blocking problem is to find a blocking in Bk such that the maximum within-block dissimilarity is minimized. This amounts to finding an optimal blocking $b^* \in B_k$ such that the largest edge cost in $G(b^*)$, is as small as possible; let λ denote this minimum:

$$\max_{ij \in E(b^*)} c_{ij} = \min_{b \in B_k} \max_{ij \in E(b)} c_{ij} \equiv \lambda \quad (4)$$

Definition 3: An α-approximation algorithm for the bottleneck threshold blocking problem derives a blocking $b \in B_k$ with a maximum within-block cost no larger than λ :

$$\max_{ij \in E(b)} c_{ij} \leq \alpha \lambda \quad (5)$$

So unless P = NP, no polynomial-time $(2 - \epsilon)$ −approximation algorithm exists for any $\epsilon > 0$. Therefore, the problem is NP-hard, and finding an optimal solution is computationally intractable except for special cases or very small samples.

2.2 Approximate of Threshold Blocking Algorithm

The threshold blocking problem can be solved with 4-approximation algorithm. The algorithm guarantees a threshold blocking with maximum within block no longer than 4λ.

$$\max_{ij \in E(b_{alg})} c_{ij} \leq 4\lambda$$

Proof: Before going into proof let's look at our lemma's

Lemma 1: For any non-seed vertex, $i \notin S$:

1. There exist no two seeds both adjacent to i in G_{nn}.
2. There exists a walk in G_{nn} of two or fewer edges from i to the seed of the block that i is assigned to.

Lemma 2: No edge cost in G_{nn} can be greater than the maximum cost in the optimal blocking

Let b_{alg} denote the blocking produced by the algorithm. Consider any within-block edge $ij \in E(b_{alg})$.We must show that c_{ij} is bounded by 4λ.

If $ij \in E_{nn}$, we have $c_{ij} \leq \lambda$ by Lemma 2. If $ij \notin E_{nn}$ and $i \notin S, j \in S$, then by Lemma 1, there exists some l so that $il, lj \in E_{nn}$. Lemma 2 applies to both these edges. By Equation 1, the triangle inequality, it follows:

$$c_{ij} \leq c_{il} + c_{lj} \leq \lambda + \lambda = 2\lambda$$

If $ij \notin E_{nn}$and $i, j \notin S$, let $l \in S$ be the seed in the block that vertices i and j are assigned to. From above we have $c_{il} + c_{lj} \leq 2\lambda$, and by the triangle inequality:

$$c_{ij} \leq c_{il} + c_{lj} \leq 2\lambda + 2\lambda = 4\lambda$$

As there is exactly one seed in each block, $i, j \in S$ is not possible and we have considered all edges in $E(b_{alg})$.

2.3 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:

Algorithm1: Threshold Blocking

1. Construct a (k-1)-nearest neighbor subgraph of G. Denote this graph $G_{nn} = (V, E_{nn})$.
2. Find a maximal independent set of vertices, S, in the second power of the (k-1)-nearest neighbor subgraph, G^2_{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 $G_{nn}, V_i = N_{G_{nn}}[i]$
4. For each yet unassigned vertex, assign it to any block that contains one of its adjacent vertices in G_{nn}.

When the algorithm terminates, the collection of blocks, $b_{alg} = \{V_i\}_{i \in S}$, is a valid threshold blocking of the experimental units that satisfies the optimality bound.

2.4 Complexity

The blocking algorithm terminates in T(n) = $O(n \log^k n)$ 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.

Chapter 3 - Literature Survey

3.1 Hierarchical Agglomerative Clustering Algorithm

This algorithm works by grouping the data one by one based on 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 based on dendogram graph we can calculate how many numbers of clusters should be actually present.

3.1.1 Algorithm

Algorithm1: HAC

```
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

Easy to implement and gives best result in some cases. In addition, No prior information about the number of clusters is required.

3.1.3 Cons

The algorithm can never undo what was done previously. The time complexity of the algorithm is at least O ($n^2 \log n$), where *'n'* is the number of data points. 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

No objective function is directly minimized. Sometimes it is difficult to identify the correct number of clusters by the dendogram.

3.1.4 Comparison with other Algorithms

Hierarchical Clustering can give different partitioning depending on the level-of-resolution we are looking at whereas K-means clustering produces a single partitioning. Hierarchical clustering does not need the number of clusters to be specified Whereas K-Means clustering needs the number of clusters to be specified. Hierarchical clustering can be slow (has to make several merge/split decisions) whereas K-means clustering is usually more efficient run-time wise

3.1.5 Complexity

For a dataset of size *n* the algorithm requires **O (n^2)** space complexity and **O (n^3)** time complexity for most of the cases but the complexity can be reduced to **O (n^2 log(n))** time for some approaches by using appropriate data structures

3.2 PAM Algorithm

PAM stands for "partitioning 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.

Algorithm1: PAM

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 x_i and selected medoid m_k 4 calculate the total swapping cost S(x_i m_k).
5 For each pair of x_i and m_k
6 If S ≤ 0,m_k is replaced by x_i
7 Assign each data item to the cluster with most similar representative item i.e. medoid.
8 Repeat steps 2-3 until there is no change in the medoids.

3.2.2 Pros

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 and it is easy to implement PAM.

3.2.3 Cons

PAM is efficient for small data sets but does not scale well for large data sets. 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 is # of clusters. 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 outliers or other extreme values less influence a medoid than a mean. However, its processing is costlier than the k-mean method.

Chapter 4 - 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 (Lamere, Million Song Dataset, 2011) . 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 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.

4.2 Data Validation

Cross-Validation is used to validate the model to check how the statistical analysis results will generalize to an independent data set. It is used here to estimate how accurately our predictive model will perform in practice. Usually in a supervised learning for a prediction problem, the known set of data (i.e., data with cluster labels) is partitioned into training data and testing data. The model is trained on training data and is validated against testing data. This is done to avoid problems like overfitting and will give an insight on how the model will generalize to an independent dataset. In case of unsupervised learning for prediction problem, the dataset does not contain labels to follow the same approach. So, the cross-validation is done against the error rate on clustering results of training data and test data. In this project, average intra-cluster distances are used to identify the differences in clustering results of training data and test data.

In the holdout method, we randomly assign data points to two sets d_0 and d_1, 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 d_0 and test on d_1. 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 DBSCAN and K-Means 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.

4.3 Feature Selection

The original Million Song dataset does not contain any labels or genre information. The goal of clustering task in the project is to predict the genre label of each song. To identify the features corresponding to the task on given data set, a feature selection algorithm PCA or Random forest is chosen. An alternative data set exists which contains partial data from the Million Song Dataset along with genre labels. This dataset is chosen to identify the predictors.

Chapter 5 - 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 inter-cluster distance, intra-cluster distance, silhouette coefficient and similarity between the cluster outputs.

In our project we will be carrying two main experiments:

1. Threshold blocking algorithm with HAC vs HAC.
2. 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 threshold blocking 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 do not 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

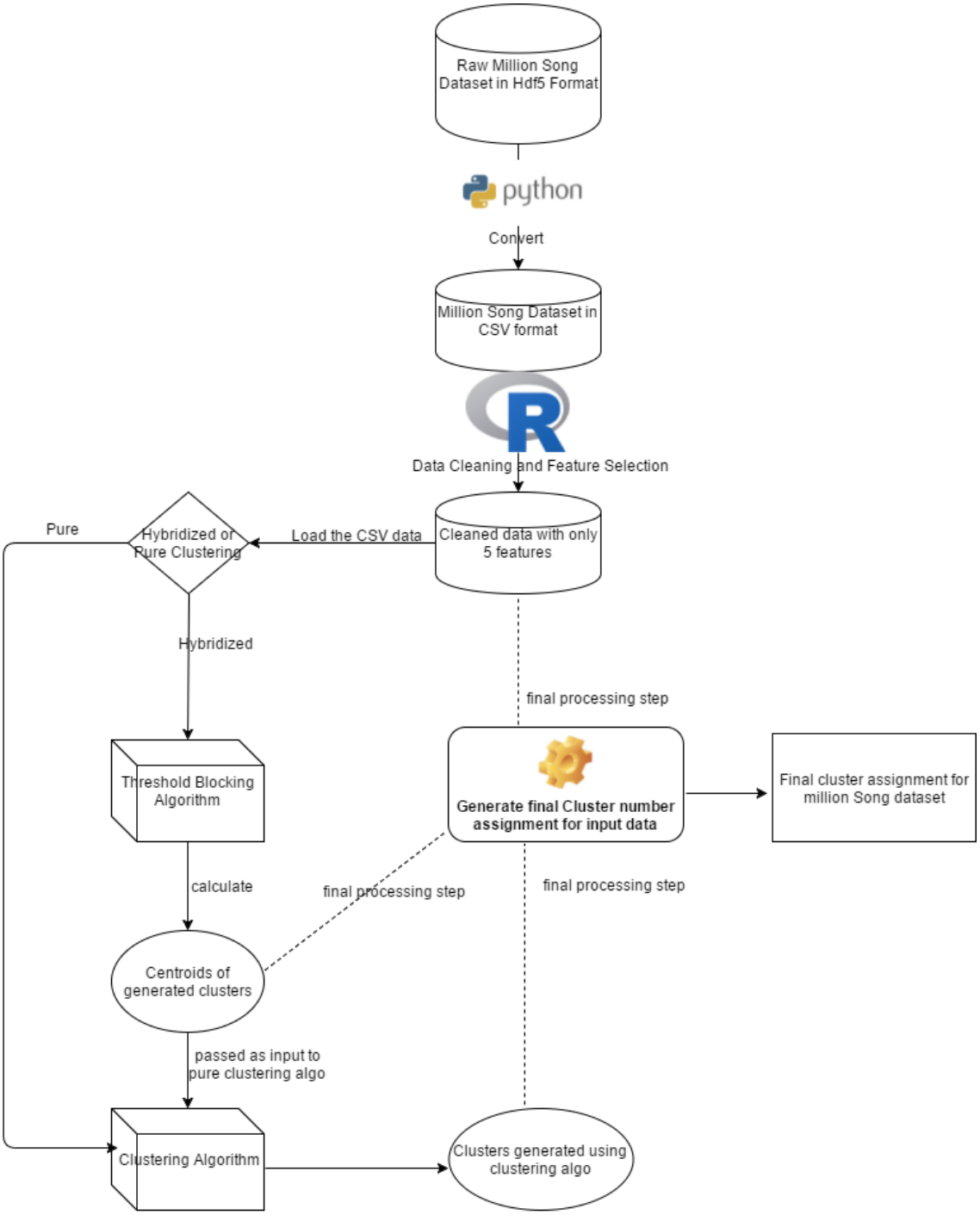

Figure 1 Data Flow Diagram for Clustering using hybridized algorithm

5.4 Implementation

1. Million Song Dataset is obtained from the following source. (Lamere, The Million Song Dataset)
2. The files provided by the dataset are in .h5 format which are converted using python wrapper code into .csv files.
3. 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.
4. Dataset consisting of Million records is used for running the hybridized HAC, hybridized PAM, HAC and PAM algorithms.
5. Due to computation limits, random subset is chosen for clustering to HAC and hybridized HAC as well as hybridized PAM and PAM.
6. 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., 30,000 record dataset for HAC and 20,000 record dataset for PAM. Given a threshold value k, the algorithm divides the dataset into blocks which consist of minimum k points.
7. 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 HAC or PAM algorithm for clustering.
8. In Experiment1, the output from threshold blocking algorithm in given to HAC algorithm for clustering. On the other part, the data set which is given to threshold blocking algorithm is given to HAC to compute the clusters for the data set. The output from hybridized HAC algorithm is compared against the output of HAC algorithm to evaluate the performance of the algorithm.
9. In Experiment 2, the output from threshold blocking algorithm in given to PAM algorithm for clustering. On the other part, the data set which is given to threshold blocking algorithm is given to PAM to compute the clusters for the data set. The output from hybridized PAM algorithm is compared against the output of PAM algorithm to evaluate the performance of the algorithm.

10. 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

No of Cores: 4

Chapter 6 - Experimental Results

Two experiments are performed on the Million Song Dataset each evaluates the performance variation in hybridized algorithm over the original algorithm for various values of k.

1) Hybridized PAM vs PAM on random subset of Million Song Dataset :-

Firstly, the threshold blocking algorithm is executed on random subset of Million Song Dataset. The centroid from the output clusters of this step is given to PAM algorithm for clustering. Thus, the centroids data set is clustered using PAM and the cluster represented by the centroid is the corresponding cluster number for the data points represented by that centroid. These two steps combinedly form the hybridized clustering algorithm. The two steps are repeated for various threshold values that is k. The original PAM algorithm that is PAM clustering algorithm is run on entire random subset of data. The threshold blocking algorithm in the previous case runs as pre-clustering step but not in this case. Also, no other pre-clustering tasks or simplifications are done on the random subset of data. The cluster output from the PAM algorithm alone is compared against each of the cluster outputs from the hybridized algorithm. This completes the first experimental setup to compare the performance of hybridized PAM with original PAM.

2) Hybridized HAC vs HAC on random subset of Million Song Dataset:-

Just like the experiment1 described above, the hybridized HAC is also executed in the similar way for various threshold values. The hybridized HAC here implies that the formation of clusters with threshold blocking algorithm on the random subset of data and clustering the centers of those clusters with HAC algorithm. Finally mapping back the cluster output of centroid to the data points represented by the centroid. Each of these clustering outputs from the hybridized HAC is evaluated against the HAC algorithm executed on the random subset of data. From the experimental observation perspective, HAC is more scalable to data when compared to hybridized HAC. This experiment is also used along with experiment1 described above to evaluate the performance of hybridized HAC vs HAC algorithm.

As mentioned above in each of these experiments in the case of hybridized algorithm, the first step is to form clusters using threshold blocking algorithm. To evaluate the performance of algorithm

the threshold blocking algorithm is planned to be executed over various values of k chosen from the range (5,10,15,20,50,60,70,80,90,100,150,200,250). But for certain values of k and for the chosen data size, the hybridized algorithm takes infinite amount of time to run. For smaller values of k, the data set is large for the algorithms to execute and for larger values of k the threshold blocking algorithm may take considerable amount of time though it will surely terminate. Also, it could be possible that from the given subset of the data, the centroids formed out of the similar units also could be very sparse enough making the cluster formation in both the algorithms difficult. Therefore, only certain k values are chosen and executed on random subset of data. The HAC or PAM each of them is run on entire subset of data and as well as on the dataset of centroids obtained from the threshold blocking algorithm output.

6.1 Cluster Evaluation Measures

(Kannamareddy, 2017) mentions the below clustering approach in her report. The same clustering measures and approaches are used in this project to evaluate the clustering outputs obtained in the experiments. 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 Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient for a sample is (b - a) / max(a, b). Where, b is the distance between a sample and the nearest cluster that the sample is not a part of. It's important to know that Silhouette Index and Silhouette Coefficient are synonyms to each other.

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 algorithm 2.

Given two clustering algorithm outputs, in an unsupervised approach the numbers from both the algorithms do not 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 cluster1 of algorithm1 present in cluster1 of algorithm2 is not correct and also leads to erroneous results. In addition, 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:

1. 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.
2. The matrix contains values of how many elements match between the cluster outputs from both the algorithms.
3. At the intersection of row *i* and column *j*, the value of the cell *ij* gives the information about how many elements of $cluster_i$ matches with elements of $cluster_j$.
4. 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.

5. 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.
6. 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.*

6.1.3.2 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

Both Hybridized HAC and HAC are executed on random subset of 1 Million Song Data set with cluster output value as 13. This cluster output value is inferred from the data set description which mentions the songs belong to 13 different genres. Since the clusters formed represent the songs with similar characteristics, all the songs belong to a genre are assumed to fall into one cluster.
The below mentioned cluster evaluation metrics and computational metrics are collected over various values of "k" to measure the performance of the algorithm. 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

The K0 in the graph below represent the silhouette coefficient for original HAC on random subset of Million Song Dataset while 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.

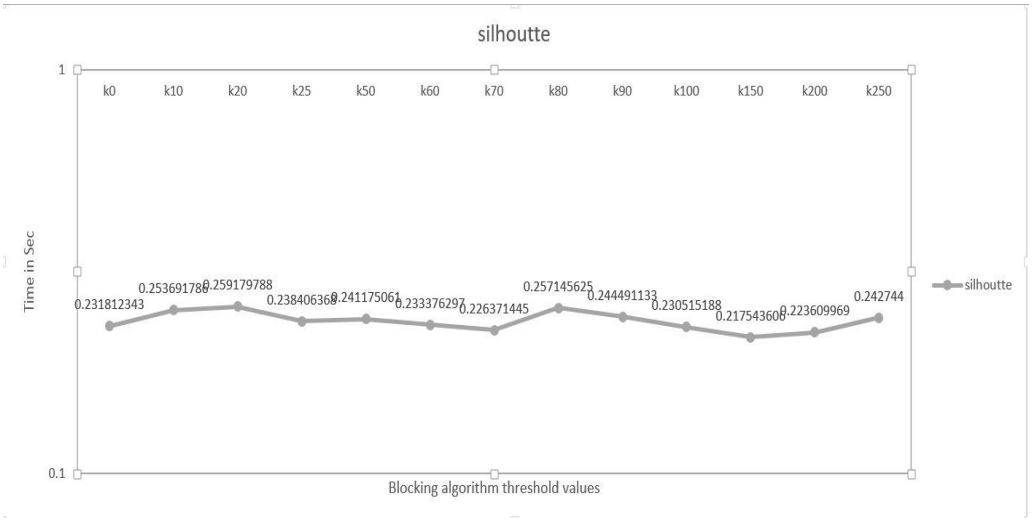

Figure 2 Silhouette Comparison for Pure HAC vs Hybridized HAC for various Threshold Values

6.2.2 Intra-Cluster Distance

The average over all the Intra-cluster distances from the 13 clusters formed is calculated and is plotted for different values of k. For k=0 which represents original HAC algorithm, the value is near 70. From the plot, it can be deduced that the average distance increases initially until threshold value k =10 which corresponds to an execution of hybridized HAC algorithm, but later decreases and remains the same having a value around 40. This indicates there is a good amount of intra-cluster similarity in the clusters obtained from the hybridized algorithm when compared to original HAC algorithm.

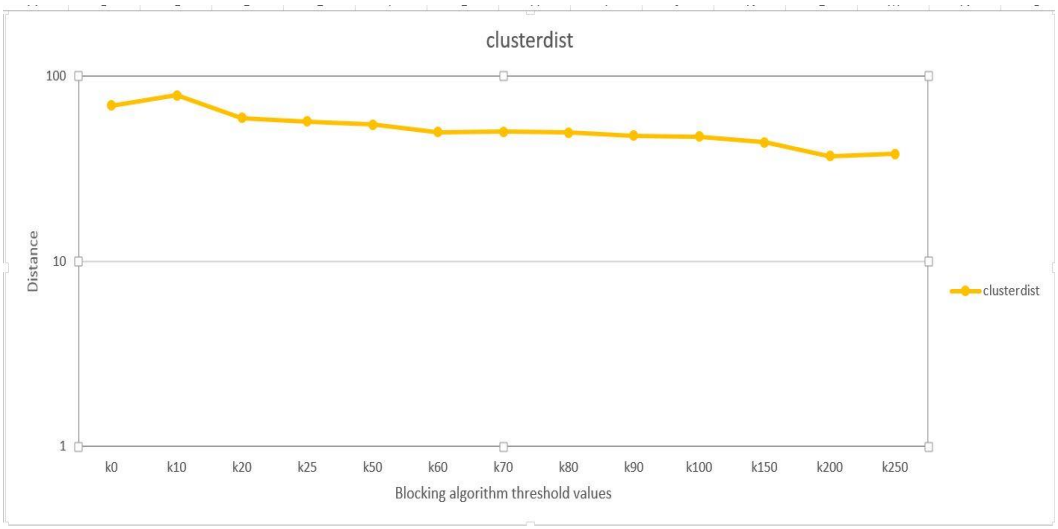

Figure 3 Intra Cluster Distance Comparison for Pure HAC vs Hybridized HAC for different threshold values

6.2.3 Inter-Cluster Distance

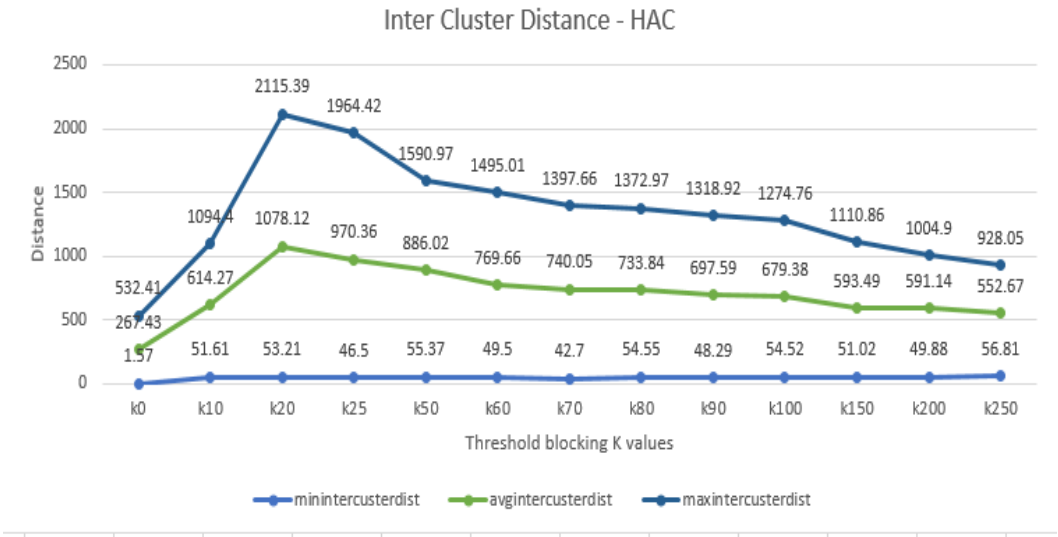

Figure 4 Inter Cluster Distance Comparison for Pure HAC vs Hybridized HAC for different threshold values

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 K=20 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.

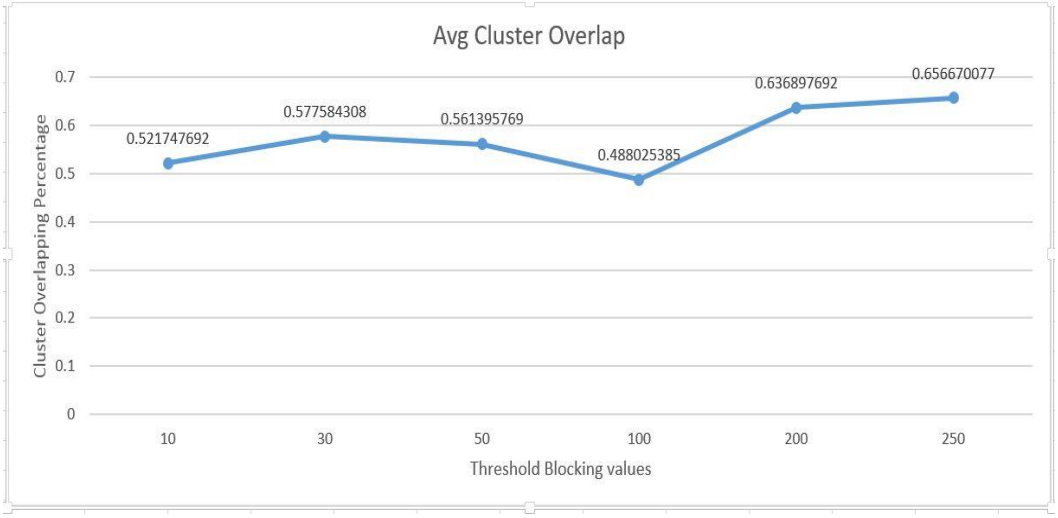

Figure 5 Cluster Overlap Percentage for Hybridized HAC with Pure HAC on different Threshold Values

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.

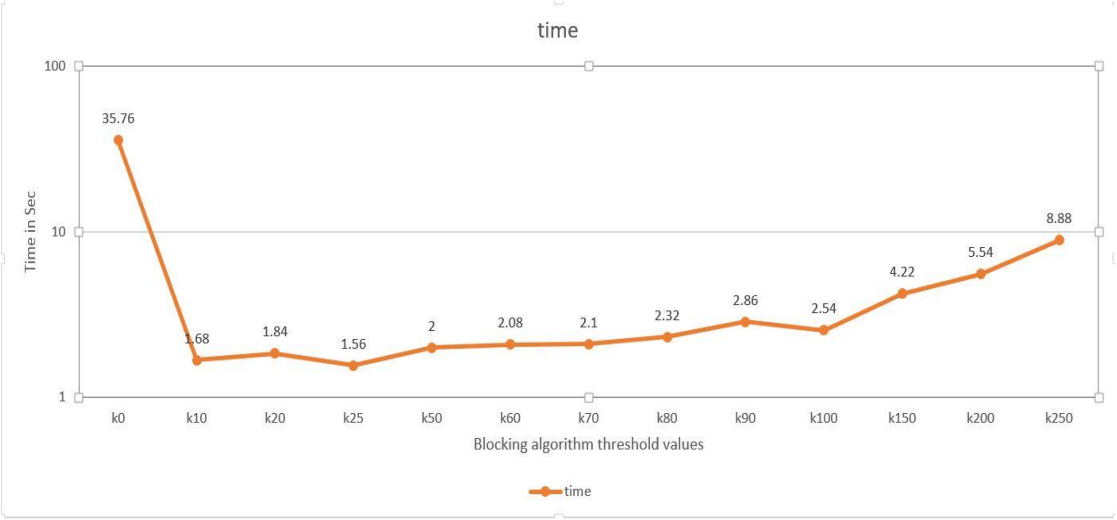

Figure 6 Processing Time Comparison for Pure HAC vs Hybridized HAC for various Threshold Values

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.

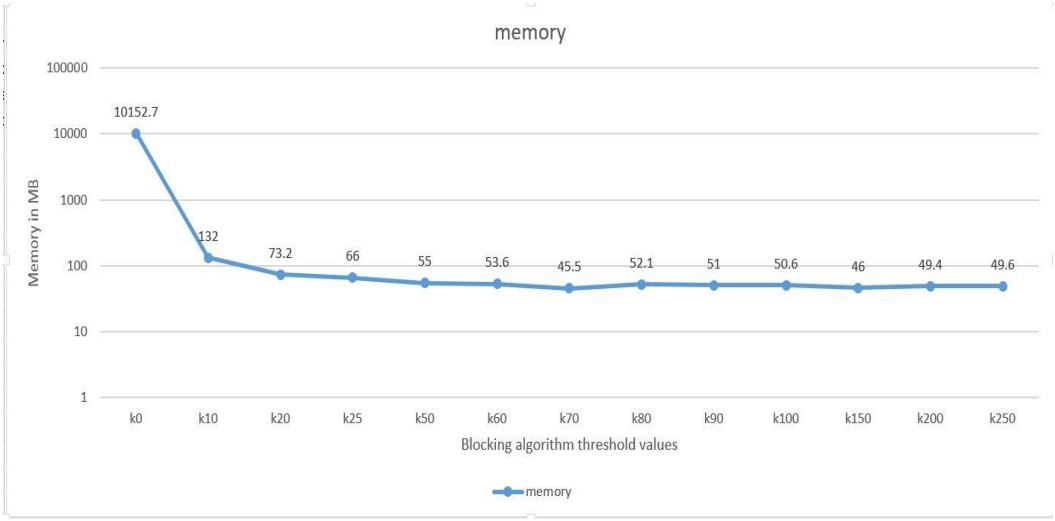

Figure 7 Memory usage Comparison for Pure HAC vs Hybridized HAC for various Threshold Values

6.3 Experiment2 - Comparing Hybridized PAM with PAM

Both Hybridized PAM and PAM are executed on random subset of 1 Million Song Data set with cluster output value as 13. This cluster output value is inferred from the data set description which mentions the songs belong to 13 different genres. Since the clusters formed represent the songs with similar characteristics, all the songs belong to a genre are assumed to fall into one cluster.

The below mentioned cluster evaluation metrics and computational metrics are collected over various values of "k" to measure the performance of the algorithm. The "k" value represents the threshold value given to threshold blocking algorithm. When k=0, it implies the data set is run on PAM alone. 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

Silhouette Coefficient values are all negative for hybridized as well as for pure PAM. But however when hybridized algorithm is used the Silhouette coefficient values improved which means that better clusters are formed when hybridized PAM is used.

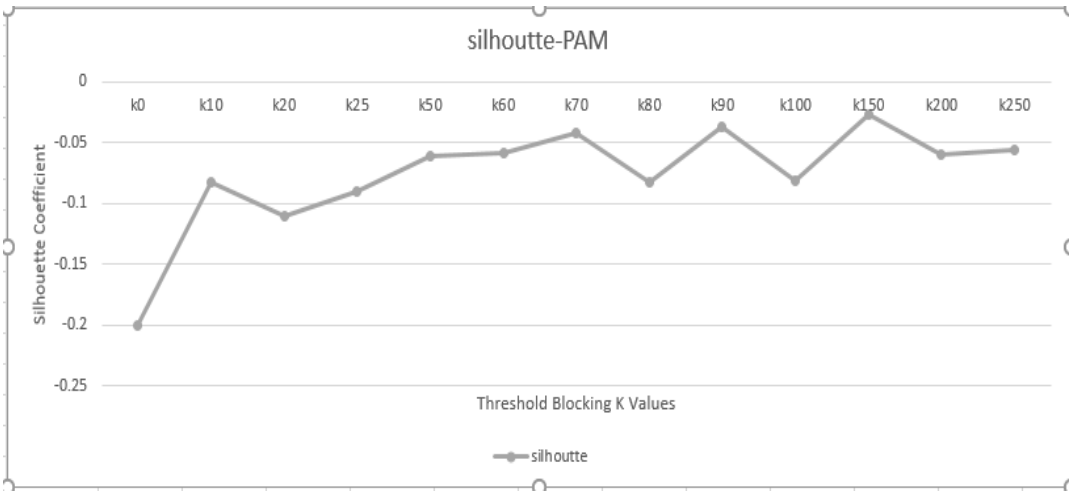

Figure 8 Silhouette Comparison for Pure PAM vs Hybridized PAM for various Threshold Values

6.3.2 Intra-Cluster Distance

Hybridized algorithm creates the clusters with less intra cluster distance when compared to clusters formed using pure algorithm. In addition, there is a drastic decrease in intra cluster distance when using hybridized algorithm when compared to pure algorithm. So it is clear that hybridized algorithm constructs better clusters compared to pure algorithm.

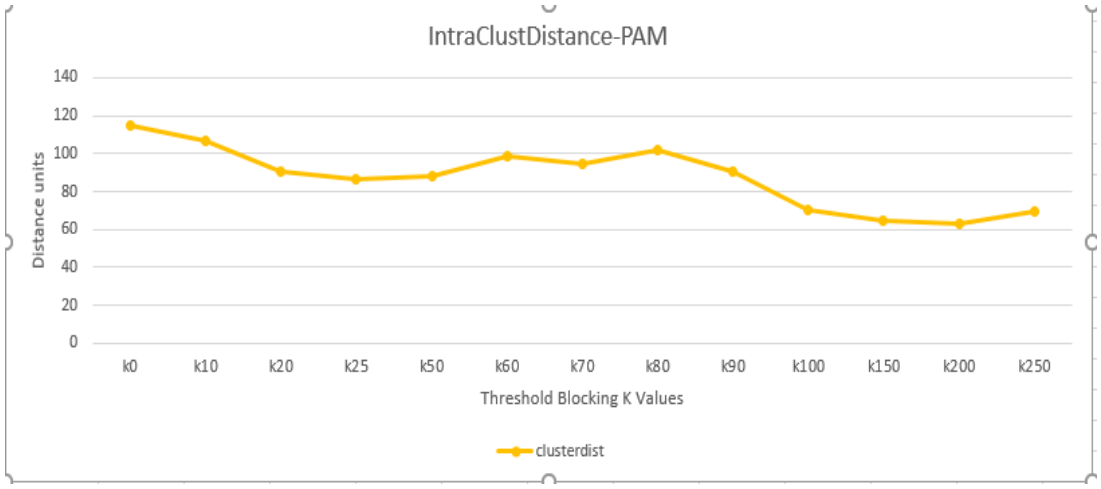

Figure 9 Intra Cluster Distance Comparison for Pure PAM vs Hybridized PAM for different threshold values

6.3.3 Inter-Cluster Distance

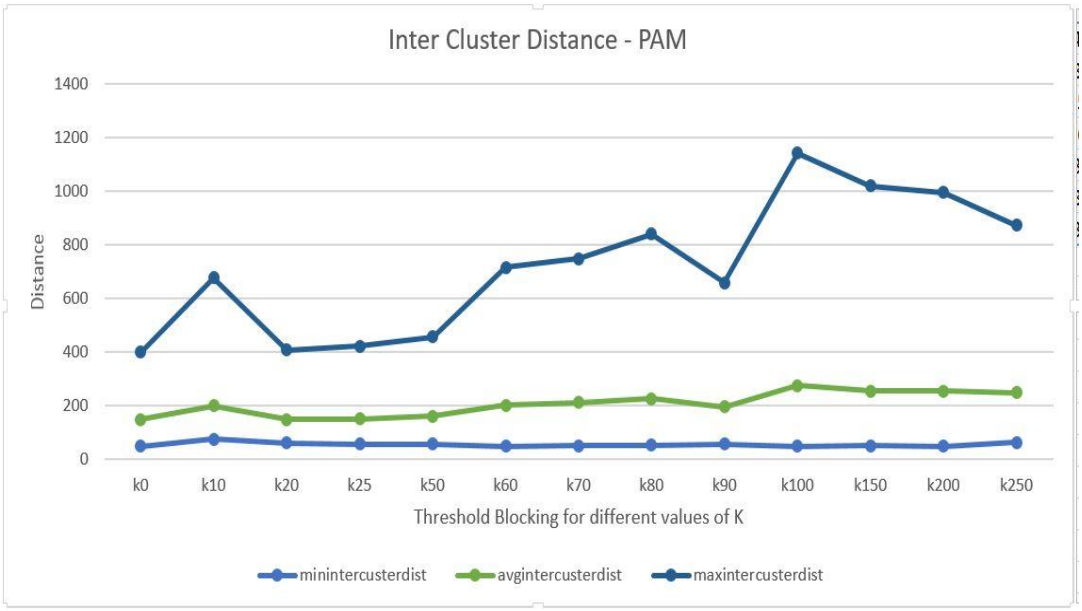

Figure 10 Inter Cluster Distance Comparison for Pure PAM vs Hybridized PAM for different threshold values

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.

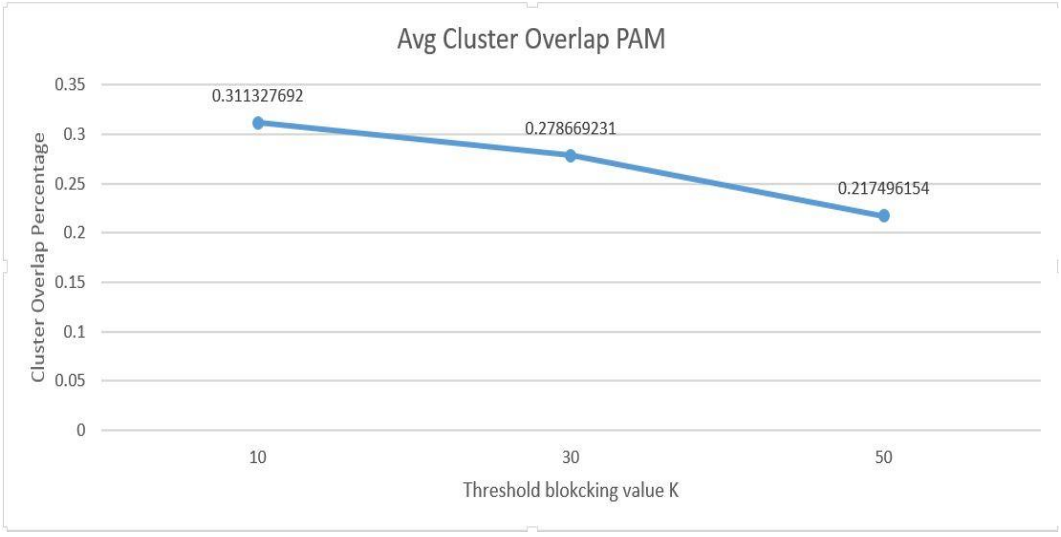

Figure 11 Cluster Overlap Percentage for Hybridized PAM with Pure PAM on different Threshold Values

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. 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.

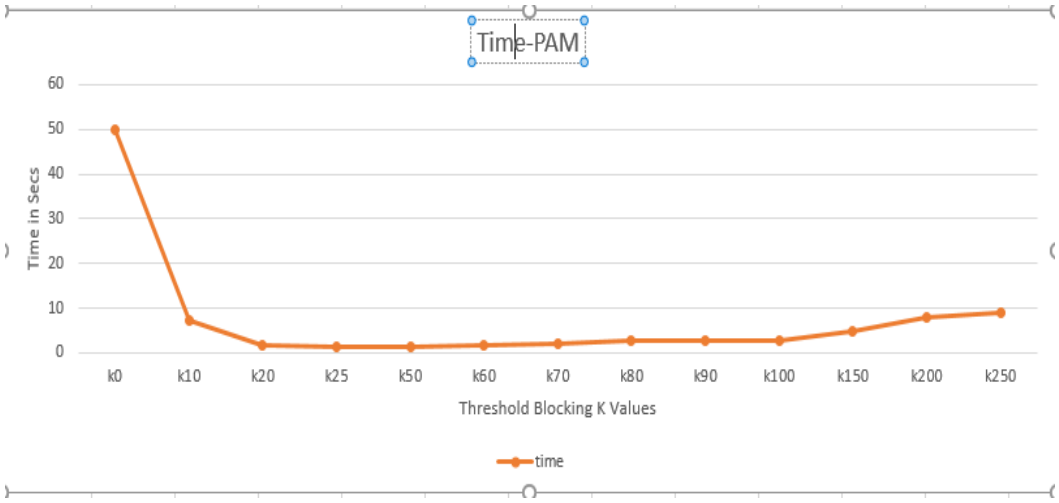

Figure 12 Processing Time Comparison for Pure PAM vs Hybridized PAM for various Threshold Values

6.3.6 Memory

Memory usage decreased with higher values of K as the number of points that are passed to PAM is reduced after preprocessing step where data size is reduced by factor of K. Therefore, it is clear that hybridized algorithms occupy less RAM Space.

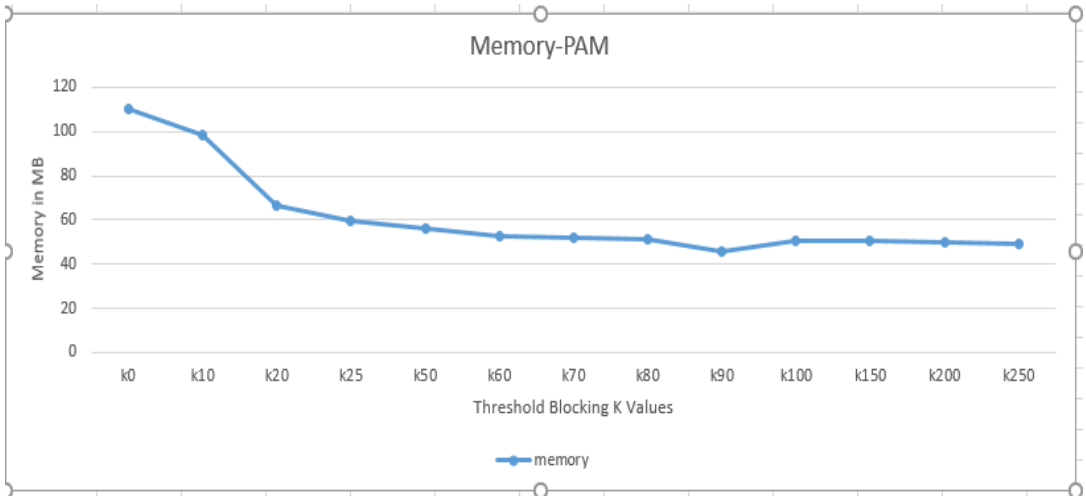

Figure 13 Memory usage Comparison for Pure PAM vs Hybridized PAM for various Threshold Values

Chapter 7 - Summary and Future Work

7.1 Summary

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 Big data 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.

7.2 Future Work

Even though the experiment helped us to understand better about the efficiency of hybridized algorithm on large dataset, the research can be expanded in many other ways. Map Reduce Framework is helpful to overcome the memory issues that will arise while calculating the clustering evaluation metrics for large datasets. In addition, threshold blocking algorithm can be executed iteratively using large and small values of k to measure the variation in performance. Further, it would be interesting to see how threshold blocking algorithm works on various data sizes.

References

HAC. (2017). Retrieved April 10, 2017, from Wikipedia: https://en.wikipedia.org/wiki/Hierarchical_clustering#Agglomerative_clustering_example

Higgins, M. J., Savje, F., & Sekhon, J. S. (2016). Improving massive experiments with threshold. *PNAS*. Retrieved April 10, 2017

Kannamareddy, A. (2017). *Density-based and partitioning-based clustering on large threshold-bounded data sets.* M.S.Report, Kansas State University. Retrieved April 10, 2017

Lamere, T. B.-M. (2011). *Million Song Dataset*. Retrieved April 10, 2017, from The Million Song Dataset: https://labrosa.ee.columbia.edu/millionsong/pages/getting-dataset

PAM. (2017). Retrieved April 10, 2017, from Wikipedia: https://en.wikipedia.org/wiki/Partitioning_Around_Medoids

Abbas, O. A. (2008). Comparisions Between Data Clustering Algorithms. *The International Arab Journal of Information Technology*. Retrieved April 10, 2017

Cavan, R., Changchun, W., & Mark, R. (2005). A RAPID METHOD FOR THE COMPARISON OF CLUSTER ANALYSES. *Statistica Sinica*, 19-33. Retrieved April 10, 2017

Cluster Analysis. (2017). Retrieved April 10, 2017, from Wikipedia: https://en.wikipedia.org/wiki/Cluster_analysis

Cluster Performance Evaluation. (2017). Retrieved April 10, 2017, from Scikit Learn: http://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

Cross Validation. (2017). Retrieved April 10, 2017, from Wikipedia: https://en.wikipedia.org/wiki/Cross-validation_(statistics)

Determining the number of clusters in a data set. (2017). Retrieved April 10, 2017, from Wikipedia: https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set

Enrique, A., Julio, G., Javier, A., & Felisa, V. (2009). A comparison of Extrinsic Clustering Evaluation Metrics based on Formal Constraints. Retrieved April 10, 2017

K-Means Clustering. (2017). Retrieved April 10, 2017, from Wikipedia: https://en.wikipedia.org/wiki/K-means_clustering

Martin, E., Hans-Peter, K., Jiirg, S., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters. *AAAI*. Retrieved April 10, 2017

R.Lleti, M.C.Ortiz, Sarabia, L., & Sanchez, M. (2004). Selecting variables for K-Means cluster analysis by using a genetic algorithm that optimises the silhouettes. *Analytica Chimica*, 87-100. Retrieved April 10, 2017

Rousseeuw, P. J. (1987). Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*, 53-65. Retrieved April 10, 2017

Silhouette Clustering. (2017). Retrieved April 10, 2017, from Wikipedia: https://en.wikipedia.org/wiki/Silhouette_(clustering)

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to Data Mining.* Retrieved April 10, 2017

Appendix A - Attributes of the 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.