Quantification of Variation of Results of the K-Means Clustering Algorithm Run Ten Times on the First Twenty-Five Primes – A Criterion for Applicability of the K-Means Clustering

Chandini Giri[1], R Satya Ravindra Babu[2]

[1] *M.Tech CSE Student, Sanketika Vidya Parishad Engineering College, Visakhapatanam, Andhra Pradesh State, India*

[2] *Associate Professor, Department of Computer Science, Sanketika Vidya Parishad Engineering College, Visakhapatanam, Andhra Pradesh State, India*

Abstract **- Clustering is one of the most common exploratory data analysis technique used to get an intuition about the structure of the data. It can be defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different. In other words, we try to find homogeneous subgroups within the data such that data points in each cluster are as similar as possible according to a similarity measure such as euclidean-based distance or correlation-based distance. The decision of which similarity measure to use is application specific. K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. K - Means clustering algorithm is a scheme for clustering continuous and numeric data. As K-Means algorithm consists of scheme of random initialization of centroids, every time it is run, it gives different or slightly different results because it may reach some local optima. Quantification of such aforementioned variation is of some importance as this sheds light on the nature of the Discrete K-Means Objective function with regards its maxima and minima. The K-Means Clustering algorithm aims at minimizing the aforementioned Objective function. In this research investigation, the author has attempted to quantify the variation of results of the K-Means Clustering Algorithm, run 10 times on the first 25 Prime numbers. Also, a notion of Percentage Uncertainty of clustering assignment for each data set point is computed for each run of the K- Means Clustering Algorithm. Also, a criterion is proposed for the applicability of K-Means Clustering Algorithm for the given data set.**

Index Terms **- K-Means Clustering, Clustering Uncertainty**

I.INTRODUCTION

[1]defines the notion of K-Means Clustering in detail. k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. It is popular for cluster analysis in data mining. k-means clustering minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. For instance, better Euclidean solutions can be found using k-medians and k-medoids.

Stuart P. Lloyd [2] advented the notion of Least squares quantization in pcm. It has long been realized that in pulse-code modulation (PCM), with a given ensemble of signals to handle, the quantum values should be spaced more closely in the voltage regions where the signal amplitude is more likely to fall. It has been shown by Panter and Dite that, in the limit as the number of quanta becomes infinite, the asymptotic fractional density of quanta per unit voltage should vary as the one-third power of the probability density per unit voltage of signal amplitudes. In this paper the corresponding result for any finite number of quanta is derived; that is, necessary conditions are found that the quanta and associated quantization intervals of an

optimum finite quantization scheme must satisfy. The optimization criterion used is that the average quantization noise power be a minimum. It is shown that the result obtained here goes over into the Panter and Dite result as the number of quanta become large. The optimum quantization schemes for 26 quanta, b = 1,2, . ,7 are given numerically for Gaussian and for Laplacian distribution of signal amplitudes.

J. MacQueen [3] details some methods for classification and analysis of multivariate observations Shenchao Du et al [4] detailed Aircraft Design Optimation with Uncertainty based On Fuzzy Clustering Analysis. According to them Uncertainty always exists in any design problems; conventional aircraft design with deterministic optimization may achieve underdesign or overdesign. Therefore, it is necessary to consider uncertainty analysis in aircraft concept design. Traditional uncertainty analyses need many sampling points to simulate the uncertain models. These methods include a large number of calculations to achieve the required accuracy. To increase the efficiency of uncertainty analysis and reduce the effect of error propagation on uncertainty models, a method with dynamic surrogate models based on fuzzy clustering analysis was proposed by them in this paper. Among the design spaces, the sampling points with little influence on response surface are abandoned by dynamic screening until the surrogate model reaches the expected level of accuracy. They then applied this method to the optimization of a hypothetical aircraft concept design, which shows that the calculated amount of uncertainty analysis can be reduced effectively while the optimized performance can satisfy the reliability and robustness.

Carl Edward Rasmussen et. al [5] carried out research on Modeling and Visualizing Uncertainty in Gene Expression Clusters Using Dirchlet Process Mixtures. Although the use of clustering methods has rapidly become one of the standard computational approaches in the literature of microarray gene expression data, little attention has been paid to uncertainty in the results obtained. Dirichlet process mixture (DPM) models provide a nonparametric Bayesian alternative to the bootstrap approach to modeling uncertainty in gene expression clustering. Most previously published applications of Bayesian model-based clustering methods have been to short time series data. In this paper, the authors presented a case study of the application of nonparametric Bayesian clustering methods to the clustering of high-dimensional non time series gene expression data using full Gaussian covariances. The authors use the probability that two genes belong to the same cluster in a DPM model as a measure of the similarity of these gene expression profiles. Conversely, this probability can be used to define a dissimilarity measure, which, for the purposes of visualization, can be input to one of the standard linkage algorithms used for hierarchical clustering. Biologically plausible results are obtained from the Rosetta compendium of expression profiles which extend previously published cluster analyses of this data.

Prasad, I.L.N. et al., [6] presented their research on Analysis of Uncertainty Inherent ti Valuation Methodologies in Construction Industry. In this research investigation, the authors presented a Scheme to analyze uncertainty inherent to valuation methodologies in the construction industry. Firstly, 63 construction projects were considered and their Uncertainities were computed for each of the valuation methodologies of Cost Approach Method, Market Approach Method and Income Approach Method. For each of the Valuation Approach, these Uncertainities are then clustered using K-Means Clustering Algorithm. Using a proposed notion of Cluster Level Uncertainty, the authors compute the Upper Bound and Lower Bound Uncertainities for the aforementioned thusly Clustered rote Uncertainities of the 63 Construction projects. Furthermore, a notion of Relative Importance Index and Ensembling Scheme is also proposed to ascribe importance coefficient to the Cluster Level Uncertainty of each Construction Project for the different valuation approaches used and combine the values of the three valuation approaches appropriately to get one value of Cluster Level Uncertainty, respectively. Cluster level Uncertainty is useful as most Construction projects have some semblance with past projects and therefore one can use the Cluster Level Uncertainty to find the Uncertainty of any Construction project in progress, i.e., which has not finished yet. For validation purposes the authors considered the above analysis for all 63 projects and repeated this scheme on the first 58 Construction Projects and for the next 5 Construction Projects, the authors used Linear Regression based Forecasting to predict the Uncertainities of the aforementioned last 5 Construction Projects. Then, the Uncertainities of the

first 58 Construction Projects and the predicted Uncertainities last 5 Construction Projects are considered and these are Clustered using K-Means Clustering Algorithm. The authors then compute the Cluster Level Uncertainities for each of the last 5 Construction Project Uncertainities using the proposed notion of Cluster Level Uncertainty and use the proposed Relative Importance Index and Ensembling Scheme to combine the values gotten by each of the three valuation approaches. Finally, the authors compared these Ensemble Values of the Validation Approach and the actual data case analysis.

Paniz Karbasi [7] presents a fast-seeding technique for K-Means Algorithm. The k-means algorithm is one of the most popular clustering techniques because of its speed and simplicity. This algorithm is very simple and easy to understand and implement. The first step of this algorithm is choosing k initial cluster centers. The way that this set of initial cluster centers are chosen, have a great effect on speed and quality of k-means. One of the most popular seeding techniques is k-means++ initialization, but this method needs k passes over the dataset. The author proposes a new seeding technique which chooses the initial centers much faster than k-means++.

[8] details the reasons for variance in the results of K-Means Clustering algorithm every time it is excuted. The output of the K-Means clustering changes from one execution to the other because of the following reasons:

KMeans is deterministic, but it depends on the initial centroids. Some methods to decide the initial centroids, such as KMeans++, have a random component. So, that is what leads to the aforesaid changes.

Finding the global minimum of a k-means clustering is NP-hard. Therefore, normally, we randomly initialize K-Means Clustering algorithm with different random seeds and use the best outcome (sub-optimal solution).

Because k_means is an unsupervised clustering method. For each execution, it does not have any pre-knowledge about the input data, so, for example, it does not know which cluster should be cluster number one, and just considers one number for each cluster during the process. But there is a solution for it, if the data is not stochastic. During all executions center of each specific cluster would not change. Hence, alongside output clustering of K-means, we can also read center of clusters that is calculated with k-means, and use them based on their distance from original coordinates, or based on their coordinates, which during all executions will not be changed, and use them to make our own unchanged clusters. For instance, we can make our own definition that cluster with the smallest coordinates should be cluster one, and so on.

This is a consequence of the random initialization of the clusters in the first iteration. To avoid different results, we should always select the same initial centroids. Selecting the optimal set of centroids is an NP-hard problem. For a better initialization it is suggested that we consider the K-Means ++ method:

Arthur, D., & Vassilvitskii, S. [9] presented k-means++ algorithm discussing the advantages of careful seeding. The k-means method is a widely used clustering technique that seeks to minimize the average squared distance between points in the same cluster. Although it offers no accuracy guarantees, its simplicity and speed are very appealing in practice. By augmenting k-means with a very simple, randomized seeding technique, the authors obtained an algorithm that is $\Theta(\log k)$-competitive with the optimal clustering. Preliminary experiments show that this augmentation improves both the speed and the accuracy of k-means, often quite dramatically.

Olivier Bachem et.al., [10] detail Distributed and Provably Good Seedings for k-Means in Constant Rounds. The k-means++ algorithm is the state of the art algorithm to solve k-Means clustering problems as the computed clusterings are $O(\log k)$ competitive in expectation. However, its seeding step requires k inherently sequential passes through the full data set making it hard to scale to massive data sets. The standard remedy is to use the k-means|| algorithm which reduces the number of sequential rounds and is thus suitable for a distributed setting. In this paper, the authors provide a novel analysis of the k-means|| algorithm that bounds the expected solution quality for any number of rounds and oversampling factors greater than k, the two parameters one needs to choose in practice. In particular, the authors show that k-means|| provides provably good clusterings even for a small, constant number of iterations. This theoretical finding explains the common observation that k-means|| performs extremely well in practice even if the number of rounds is low. The authors further provide a hard instance that shows that an additive error term

as encountered in this analysis is inevitable if less than k−1 rounds are employed.

Olivier Bachem et.al., [11] discuss in detail about Fast and Provably Good Seedings for k-Means. Seeding – the task of finding initial cluster centers – is critical in obtaining high quality clusterings for k-Means. However, k-means++ seeding, the state-of-the-art algorithm, does not scale well to massive datasets as it is inherently sequential and requires k full passes through the data. It was recently shown that Markov chain Monte Carlo sampling can be used to efficiently approximate the seeding step of k-means++. However, this result requires assumptions on the data generating distribution. The authors propose a simple yet fast seeding algorithm that produces provably good clusterings even without assumptions on the data. The authors analysis shows that the algorithm allows for a favourable trade-off between solution quality and computational cost, speeding up k-means++ seeding by up to several orders of magnitude. The authors validate their theoretical results in extensive experiments on a variety of real-world data sets.

Fouad Khan [12] presented his research on An Initial Seed Selection Algorithm for K-means Clustering of Georeferenced Data to Improve Replicability of Cluster Assignments for Mapping Application. K-means is one of the most widely used clustering algorithms in various disciplines, especially for large datasets. However, the method is known to be highly sensitive to initial seed selection of cluster centers. K-means++ has been proposed to overcome this problem and has been shown to have better accuracy and computational efficiency than k-means. In many clustering problems though –such as when classifying georeferenced data for mapping applications- standardization of clustering methodology, specifically, the ability to arrive at the same cluster assignment for every run of the method i.e. replicability of the methodology, may be of greater significance than any perceived measure of accuracy, especially when the solution is known to be non-unique, as in the case of k-means clustering. The author proposes a simple initial seed selection algorithm for k-means clustering along one attribute that draws initial cluster boundaries along the "deepest valleys" or greatest gaps in dataset. Thus, it incorporates a measure to maximize distance between consecutive cluster centers which augments the conventional k-means optimization for minimum distance between cluster center and cluster members. Unlike existing initialization methods, no additional parameters or degrees of freedom are introduced to the clustering algorithm. This improves the replicability of cluster assignments by as much as 100% over k-means and k-means++, virtually reducing the variance over different runs to zero, without introducing any additional parameters to the clustering process. Further, the proposed method is more computationally efficient than k-means++ and in some cases, more accurate.

K. Karteeka Pavan et.al., [13] carried out research on Robust seed selection algorithm for k-means type algorithms - Optimal centroids using high density object. Selection of initial seeds greatly affects the quality of the clusters and in k-means type algorithms. Most of the seed selection methods result different results in different independent runs. The authors propose a single, optimal, outlier insensitive seed selection algorithm for k-means type algorithms as extension to k-means++. The experimental results on synthetic, real and on microarray data sets demonstrated that effectiveness of the new algorithm in producing the clustering results.

II PROBLEM STATEMENT

The various steps of the problem of concern are:

1. Consider the first 25 Prime numbers starting with 2 as the first prime.
2. Perform K-Means Clustering Algorithm on the data of the first 25 primes (aforementioned) 10 times.
3. Ascertain Cluster Assignments of the data in every run of the 10 runs of the K-Means Clustering Algorithm.
4. Evaluate Cluster Centroids of Each Cluster for every run of the 10 runs of the K-Means Clustering Algorithm.
5. Coin a definition of Uncertainty of the Cluster Assignments from the data of the 10 Cluster Assignments (varying) gotten by 10 runs of the K-Means Clustering Algorithm.
6. Using this definition we compute percentage uncertainty for each data point for each run of the 10 runs of the K-Means Clustering Algorithm.
7. Finally plotting the percentage uncertainty for each data point for each run of the 10 runs of the K-Means Clustering Algorithm.

8. Propose a criterion is proposed for the applicability of K-Means Clustering Algorithm for the given data set.
9. Test whether K-Means Clustering Algorithm can be applied on the considered data set based on the criterion stated in 8.

III EXISTING THEORY

K- Means Clustering Algorithm

Clustering is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often according to some defined distance measure.

K- Means method falls in the category of Partitional Clustering.

Common Distance measures

Distance measure will determine how the similarity of two elements is calculated and it will influence the shape of the clusters.

They include:

1. The Euclidean distance (also called 2-norm distance) is given by:

$$d(x,y)=\left(\sum_{i=1}^{n}\left(|x_i-y_i|\right)^2\right)^{1/2}$$

2. The Manhattan distance (also called taxicab norm or 1-norm) is given by:

$$d(x,y)=\sum_{i=1}^{n}|x_i-y_i|$$

3. Minkowski Distance

$$d(x,y)=\left(\sum_{i=1}^{n}\left(|x_i-y_i|\right)^p\right)^{1/p}$$

4. Inner product space: The angle between two vectors can be used as a distance measure when clustering high dimensional data.

a. Unnormalized

$$d(x,y)=\sum_{i=1}^{n}(x_i y_i)$$

b. Normalized

$$d(x,y)=\sum_{i=1}^{n}\left\{\left(\frac{x_i}{\sqrt{\sum_{i=1}^{n}x_i^2}}\right)\left(\frac{y_i}{\sqrt{\sum_{i=1}^{n}y_i^2}}\right)\right\}$$

The k-means algorithm is an algorithm to cluster m objects based on attributes into *K* partitions, where $K<m$.

It assumes that the object attributes form a vector space.

An algorithm for partitioning (or clustering) N data points into K disjoint subsets S_j containing data points so as to minimize the sum-of-squares criterion.

$$J=\sum_{j=1}^{K}\sum_{m\in S_j}\left|x_m-\mu_j\right|^2$$

where x_m is a vector representing the the m^{th} data point and μ_j is the geometric centroid of the data points in S_j.

Simply speaking K-means clustering is an algorithm to classify or to group the objects based on attributes/features into K number of groups. K is positive integer number.

The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

How the K-Mean Clustering algorithm works?

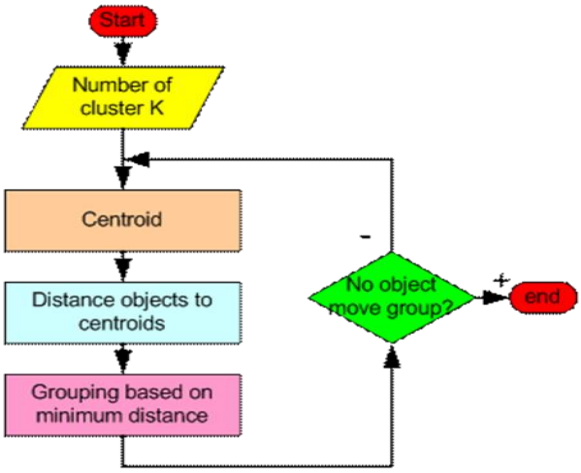

Fig 1- Flow chart showing the working of the K-Means Clustering Algorithm

Step 1: Begin with a decision on the value of K = number of clusters.

Step 2: Put any initial partition that classifies the data into K clusters. You may assign the training samples

randomly, or systematically as the following: Take the first k training sample as single element clusters

Assign each of the remaining ($m-K$) training sample to the cluster with the nearest centroid. After each assignment, recompute the centroid of the gaining cluster.

Step 3: Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.

Step 4. Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

Choosing the right number (K) of Clusters

The Elbow Method First of all, we compute the sum of squared error (SSE) for some values of K (for example 2, 4, 6, 8, etc.). The SSE is defined as the sum of the squared distance between each member of the cluster and its centroid. Mathematically, it is J as defined already. If we plot K against the SSE, we will see that the error decreases as K gets larger; this is because when the number of clusters increases, they should be smaller, so distortion is also smaller. The idea of the elbow method is to choose the K at which the SSE decreases abruptly. This produces an "elbow effect" in the graph of K against SSE.

Cluster Evaluation-Silhouette Score

The Silhouette Score is a measure of how much similarity an object bears to its own cluster (cohesion) compared to other clusters (separation). The values of the Silhouette Score range from -1 to +1. When the Silhouette Score is high, it indicates how well an object matches to its own cluster and how poorly it matches with the neighbouring clusters.

In our study, we calculate the Silhouette Score in the Euclidean Distance Metric.

Firstly, we compute the mean distance between $i \in C_i$ (data point i in the cluster C_i) and all other data points in the same cluster, as

$$a(i) = \frac{1}{|C_i|-1} \sum_{j \in C_i, i \neq j} d(i, j)$$

where $d(i, j)$ is the distance between data points i and j in the cluster C_i and $|C_i|$ indicates the number of data points in the Cluster C_i. We divide by $|C_i|-1$ as we do not include the distance $d(i,i)$ in the sum. The value $a(i)$ can be interpreted as a measure of how well i belongs to its cluster (the smaller the value, the better the belongingness).

We now compute the mean distance of point i to some cluster C_k as the mean of the distance from i to all points in C_k. That is, we compute $\frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$

For each data point $i \in C_i$, we define.

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

to be the smallest mean distance of i to all points in any other cluster, and the cluster with this smallest aforementioned mean distance is said to be the neighbouring cluster of i.

The Silhouette Score of one data point i is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \text{ if } |C_1| > 1 \text{ and}$$

$$s(i) = 0 \text{, if } |C_1| = 1$$

Existing Definition of Uncertainty of K-Means Clustering

[6] presents in detail a notion of definition of Uncertainty of K-Means Clsutering. It is detailed as follows:

Cluster Level Uncertainty

Lower & Upper Bound Uncertainty implies that each point of p^{th} Cluster, i.e., C_p has an Uncertainty of

$$\left(Uncertainity(j)_i - \min_{all\ i \in C_p} (Uncertainity(j)_i) \right)$$

on the Lower side and

$$\left(\max_{all\ i \in C_p} (Uncertainity(j)_i) - Uncertainity(j)_i \right)$$

on the Upper Side, when the Uncertainty points

(representing the various feature data points) are clustered using K-Means Clustering Algorithm.

Here, $Uncertainity(j)_i$ is the value of the Uncertainty of the i^{th} feature data point computed using j^{th} approach, $\underset{all\ i \in C_p}{Min}(Uncertainity(j)_i)$ is the Minimum value of the Cluster C_p. $\underset{all\ i \in C_p}{Max}(Uncertainity(j)_i)$ is the Maximum value of value of the Cluster C_p.

$$LBU = \left(Uncertainity(j)_i - \underset{all\ i \in C_p}{Min}(Uncertainity(j)_i)\right)$$

$$UBU = \left(\underset{all\ i \in C_p}{Max}(Uncertainity(j)_i) - Uncertainity(j)_i\right)$$

This uncertainty is the Cluster Level Uncertainty of the point after the points have been clustered using K-Means Algorithm. This macro group level uncertainty is useful because it represents the uncertainty of a feature data point with respects to all the points of the cluster or group to which it belongs wherein these points resemble each other more than the points outside of the Cluster.

IV PROPOSED THEORY

Definition of Uncertainty of K-Means Clustering

We propose the following definition of Uncertainty of K-Means Clustering:

$$\%\ Uncertainity(x_i)_j = 100\left\{\frac{\mu_i - \bar{x}_{ij}}{\mu_i}\right\}$$

where x_i is the i^{th} data point of the data set considered on which we perform K-Means Clustering (Algorithm),

$\%\ Uncertainity(x_i)_j$ is the % Uncertainty of the i^{th} data point at the j^{th} run of the K-Means Clustering Algorithm,

μ_i is the Average of the Centroids of the Clusters to which the data point x_i belonged to in the 10 runs of the K-Means Clustering Algorithm

and $\bar{x}_{ij}$ is the Centroid of the Cluster to which the data point x_i belongs to in the j^{th} run of the K-Means Clustering Algorithm.

Criterion for Applicability of the K-Means Clustering Algorithm

We compute a value given by

$$s = \left\{\frac{\left(\sum_{i=1}^{n} x_i\right)}{n}\right\}$$

where x_i is the i^{th} data point,

n is the number of data points of the data set considered on which we perform K-Means Clustering (Algorithm)

We now say,

$$\left\{\frac{s - x_i}{s}\right\}100 > \left\{\frac{\sum_{j=1}^{w}\{\%\ Uncertainity(x_i)_j\}}{w}\right\},$$

then the data point x_i has passed the K-Means applicability criterion. Here, w is the number of runs of the K-Means Clustering Algorithm.

Similarly, if $\left\{\frac{\sum_{i=1}^{n}\left\{\frac{s-x_i}{s}\right\}100}{n}\right\} > \left\{\frac{\sum_{i=1}^{n}\left|\frac{\sum_{j=1}^{w}\{\%\ Uncertainty(x_i)_j\}}{w}\right|}{n}\right\}$

, then K-Means Clustering Algorithm can be said applicable on the considered data set.

Note of Insight

It should be noted that here, in our study, we have considered 10 runs of which only distinct random centroid initialization happened only 3 times. And

$${}^{n}C_K = \frac{n!}{K!(n-K)!}$$

there exist number of possible distinct cases available for random centroid initializations, K being the number of Clusters considered. Hence, if we wish to have very reliable applicability criterion, we need to consider large number of runs of the K-Means Clustering Algorithm.

V RESULTS & CONCLUSIONS

The results and conclusions are detailed as follows:

Table 1: Cluster Assignments

		Cluster Assignments									
Sl. No	*Prime Number*	*Run 1*	*Run 2*	*Run 3*	*Run 4*	*Run 5*	*Run 6*	*Run 7*	*Run 8*	*Run 9*	*Run 10*
1	2	2	5	3	3	5	3	1	4	2	1
2	3	2	5	3	3	5	3	1	4	2	1
3	5	2	5	3	3	5	3	1	4	2	1
4	7	2	5	3	3	5	3	1	4	2	1
5	11	2	5	3	3	5	3	1	4	2	1
6	13	2	5	3	3	5	3	1	4	2	1
7	17	1	4	3	1	1	1	5	3	2	5
8	19	1	4	2	1	1	1	5	3	2	5
9	23	1	4	2	1	1	1	5	3	2	5
10	29	1	4	2	1	1	1	5	3	5	5
11	31	1	4	2	1	1	1	5	3	5	5
12	37	3	2	2	2	3	4	2	1	5	2
13	41	3	2	2	2	3	4	2	1	5	2
14	43	3	2	2	2	3	4	2	1	5	2
15	47	3	2	1	2	3	4	2	1	5	2
16	53	3	2	1	2	3	4	2	2	3	2
17	59	4	3	1	5	2	2	3	2	3	3
18	61	4	3	1	5	2	2	3	2	3	3
19	67	4	3	5	5	2	2	3	2	3	3
20	71	4	3	5	5	2	2	3	2	4	3
21	73	4	3	5	5	2	2	3	2	4	3
22	79	5	1	5	4	4	5	4	5	4	4
23	83	5	1	4	4	4	5	4	5	4	4
24	89	5	1	4	4	4	5	4	5	1	4
25	97	5	1	4	4	4	5	4	5	1	4

Table 2 – Cluster Centroids

		Cluster Centroids										
Sl. No	Prime Number	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	Average
1	2	6.833	6.833	9.625	6.833	6.833	6.833	6.833	6.833	11.11	9.625	7.8191
2	3	6.833	6.833	9.625	6.833	6.833	6.833	6.833	6.833	11.11	9.625	7.8191
3	5	6.833	6.833	9.625	6.833	6.833	6.833	6.833	6.833	11.11	9.625	7.8191
4	7	6.833	6.833	9.625	6.833	6.833	6.833	6.833	6.833	11.11	9.625	7.8191
5	11	6.833	6.833	9.625	6.833	6.833	6.833	6.833	6.833	11.11	9.625	7.8191
6	13	6.833	6.833	9.625	6.833	6.833	6.833	6.833	6.833	11.11	9.625	7.8191
7	17	23.8	23.8	9.625	23.8	23.8	23.8	23.8	23.8	11.11	9.625	19.696
8	19	23.8	23.8	9.625	23.8	23.8	23.8	23.8	23.8	11.11	9.625	19.696
9	23	23.8	23.8	34	23.8	23.8	23.8	23.8	23.8	11.11	34	24.571
10	29	23.8	23.8	34	23.8	23.8	23.8	23.8	23.8	38	34	27.26
11	31	23.8	23.8	34	23.8	23.8	23.8	23.8	23.8	38	34	27.26
12	37	44.2	44.2	34	44.2	44.2	44.2	44.2	44.2	38	34	41.54
13	41	44.2	44.2	34	44.2	44.2	44.2	44.2	44.2	38	34	41.54
14	43	44.2	44.2	34	44.2	44.2	44.2	44.2	44.2	38	34	41.54
15	47	44.2	44.2	55	44.2	44.2	44.2	44.2	44.2	38	55	45.74
16	53	44.2	44.2	55	44.2	44.2	44.2	44.2	44.2	60	55	47.94
17	59	66.2	66.2	55	66.2	66.2	66.2	66.2	66.2	60	55	63.34
18	61	66.2	66.2	55	66.2	66.2	66.2	66.2	66.2	60	55	63.34
19	67	66.2	66.2	72.5	66.2	66.2	66.2	66.2	66.2	60	72.5	66.84
20	71	66.2	66.2	72.5	66.2	66.2	66.2	66.2	66.2	76.5	72.5	68.49
21	73	66.2	66.2	72.5	66.2	66.2	66.2	66.2	66.2	76.5	72.5	68.49
22	79	87	87	72.5	87	87	87	87	87	76.5	72.5	83.05
23	83	87	87	89.666	87	87	87	87	87	76.5	89.666	86.4832
24	89	87	87	89.666	87	87	87	87	87	93	89.666	88.1332
25	97	87	87	89.666	87	87	87	87	87	93	89.666	88.1332

Table 3 – Percentage Uncertainties of Clustering Assignments of The First 25 Primes Over 10 Runs of the K-Means Clustering Algorithm

Percentage Uncertainty of The First 25 Primes Listed Vertically for the 10 Runs of the K-Means Clustering Algorithm										
Average	% U Run 1	% U Run 2	% U Run 3	% U Run 4	% U Run 5	% U Run 6	% U Run 7	% U Run 8	% U Run 9	% U Run 10
7.8191	12.61143	12.61143	-23.096	12.61143	12.61143	12.61143	12.61143	12.61143	-42.088	-23.096
7.8191	12.61143	12.61143	-23.096	12.61143	12.61143	12.61143	12.61143	12.61143	-42.088	-23.096
7.8191	12.61143	12.61143	-23.096	12.61143	12.61143	12.61143	12.61143	12.61143	-42.088	-23.096
7.8191	12.61143	12.61143	-23.096	12.61143	12.61143	12.61143	12.61143	12.61143	-42.088	-23.096
7.8191	12.61143	12.61143	-23.096	12.61143	12.61143	12.61143	12.61143	12.61143	-42.088	-23.096
7.8191	12.61143	12.61143	-23.096	12.61143	12.61143	12.61143	12.61143	12.61143	-42.088	-23.096
19.696	-20.8367	-20.8367	51.13221	-20.8367	-20.8367	-20.8367	-20.8367	-20.8367	43.59261	51.13221
19.696	-20.8367	-20.8367	51.13221	-20.8367	-20.8367	-20.8367	-20.8367	-20.8367	43.59261	51.13221
24.571	3.137845	3.137845	-38.3745	3.137845	3.137845	3.137845	3.137845	3.137845	54.7841	-38.3745
27.26	12.69259	12.69259	-24.7249	12.69259	12.69259	12.69259	12.69259	12.69259	-39.3984	-24.7249
27.26	12.69259	12.69259	-24.7249	12.69259	12.69259	12.69259	12.69259	12.69259	-39.3984	-24.7249
41.54	-6.40347	-6.40347	18.15118	-6.40347	-6.40347	-6.40347	-6.40347	-6.40347	8.521907	18.15118
41.54	-6.40347	-6.40347	18.15118	-6.40347	-6.40347	-6.40347	-6.40347	-6.40347	8.521907	18.15118
41.54	-6.40347	-6.40347	18.15118	-6.40347	-6.40347	-6.40347	-6.40347	-6.40347	8.521907	18.15118
45.74	3.366856	3.366856	-20.2449	3.366856	3.366856	3.366856	3.366856	3.366856	16.92173	-20.2449
47.94	7.801418	7.801418	-14.7267	7.801418	7.801418	7.801418	7.801418	7.801418	-25.1564	-14.7267
63.34	-4.51531	-4.51531	13.16704	-4.51531	-4.51531	-4.51531	-4.51531	-4.51531	5.273129	13.16704
63.34	-4.51531	-4.51531	13.16704	-4.51531	-4.51531	-4.51531	-4.51531	-4.51531	5.273129	13.16704
66.84	0.95751	0.95751	-8.46798	0.95751	0.95751	0.95751	0.95751	0.95751	10.23339	-8.46798
68.49	3.343554	3.343554	-5.85487	3.343554	3.343554	3.343554	3.343554	3.343554	-11.6951	-5.85487
68.49	3.343554	3.343554	-5.85487	3.343554	3.343554	3.343554	3.343554	3.343554	-11.6951	-5.85487
83.05	-4.75617	-4.75617	12.70319	-4.75617	-4.75617	-4.75617	-4.75617	-4.75617	7.886815	12.70319
86.4832	-0.59757	-0.59757	-3.68025	-0.59757	-0.59757	-0.59757	-0.59757	-0.59757	11.54351	-3.68025
88.1332	1.285781	1.285781	-1.73919	1.285781	1.285781	1.285781	1.285781	1.285781	-5.5221	-1.73919
88.1332	1.285781	1.285781	-1.73919	1.285781	1.285781	1.285781	1.285781	1.285781	-5.5221	-1.73919

Table 4 – Silhouette Widths of Clusters Gotten by the K-Means Clustering Algorithm Runs

Silhouette Widths (Cluster Evaluation)						
Run Instance	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Average
1	0.4187	0.6339	0.5012	0.5144	0.4829	0.5163
2	0.4861	0.5012	0.5164	0.4187	0.6339	0.5172
3	0.4301	0.3795	0.6365	0.4015	0.5064	0.4928
4	0.4208	0.5042	0.6372	0.4829	0.5183	0.5188
5	0.4371	0.5164	0.5012	0.4861	0.6488	0.5244
6	0.4260	0.5210	0.6372	0.5062	0.4912	0.5222
7	0.6488	0.5043	0.5144	0.4829	0.4408	0.5248
8	0.5012	0.5196	0.4187	0.6339	0.4912	0.5186
9	0.5020	0.6153	0.4424	0.4028	0.5018	0.5173
10	0.6444	0.4223	0.4957	0.4015	0.3975	0.4967

For our data set, we have LHS = $\left\{\frac{\sum_{i=1}^{n}\left\{\frac{s-x_i}{s}\right\}100}{n}\right\}$ = 42.3999 < RHS = $\left\{\frac{\sum_{i=1}^{n}\left|\left\{\frac{\sum_{j=1}^{w}\{\%\ Uncertainity(x_i)_j\}}{w}\right\}\right|}{n}\right\}$ =59.0943, hence as per the proposed criterion we cannot apply K-Means Clustering Algorithm on the considered data set for best results.

Percentage Uncertainty Plots

The following are the Percentage Uncertainty Plots of Clustering Assignments of The First 25 Primes Over 10 Runs of the K-Means Clustering Algorithm:

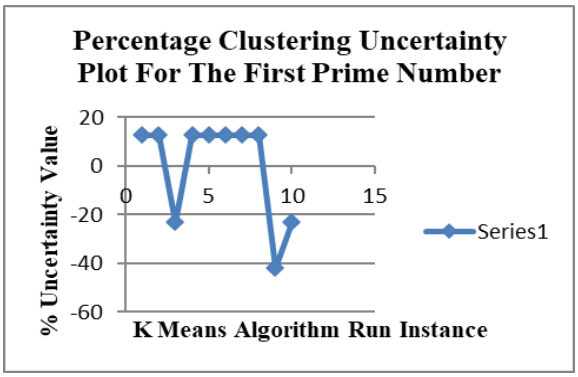

Fig 2 - Percentage Clustering Uncertainty Plot for the First Prime Number

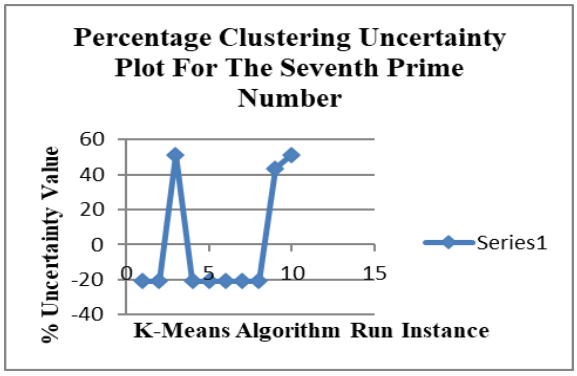

Fig 3 - Percentage Clustering Uncertainty Plot for the Seventh Prime Number

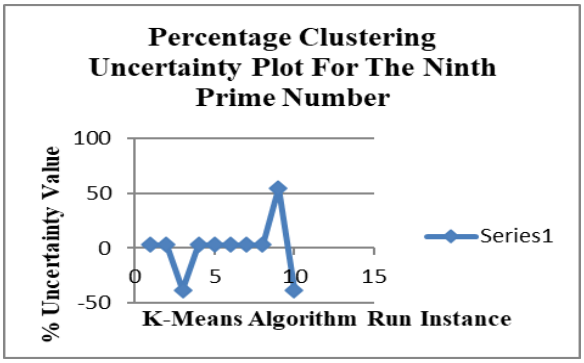

Fig 4 - Percentage Clustering Uncertainty Plot for the Ninth Prime Number

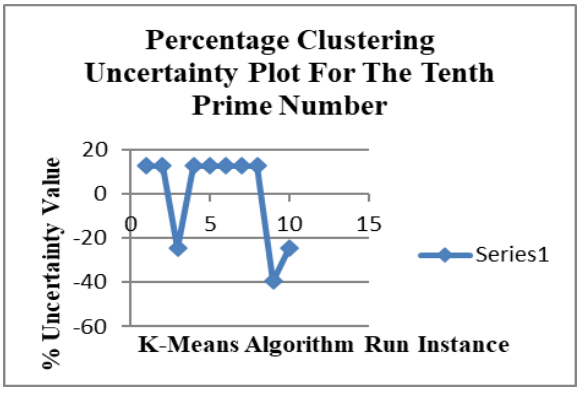

Fig 5 - Percentage Clustering Uncertainty Plot for the Tenth Prime Number

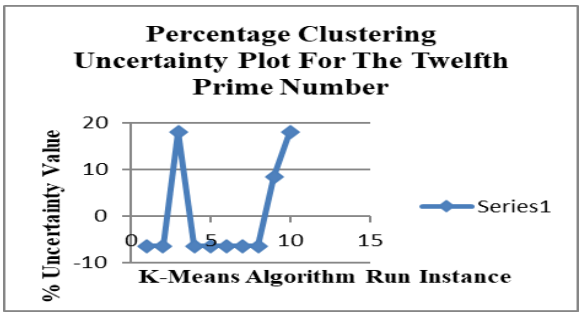

Fig 6 - Percentage Clustering Uncertainty Plot for the Twelfth Prime Number

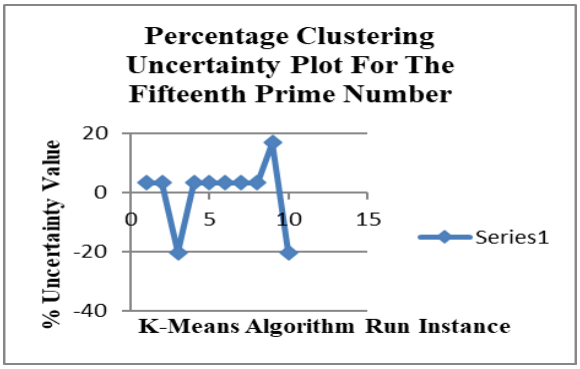

Fig 7 - Percentage Clustering Uncertainty Plot for the Fifteenth Prime Number

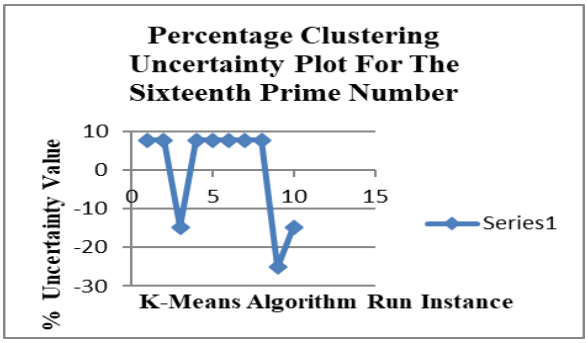

Fig 8 - Percentage Clustering Uncertainty Plot For The Sixteenth Prime Number

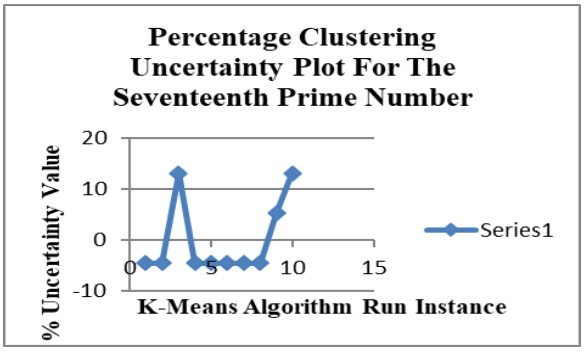

Fig 9 - Percentage Clustering Uncertainty Plot for the Seventeenth Prime Number

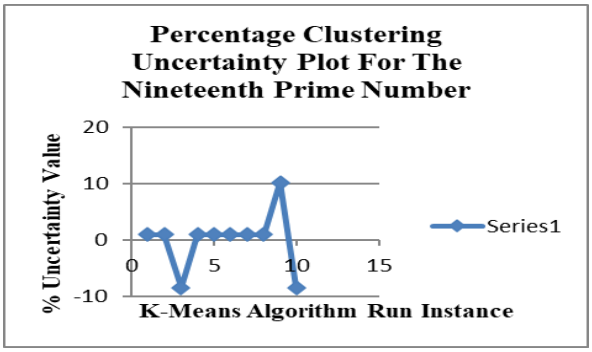

Fig 10 - Percentage Clustering Uncertainty Plot for the Nineteenth Prime Number

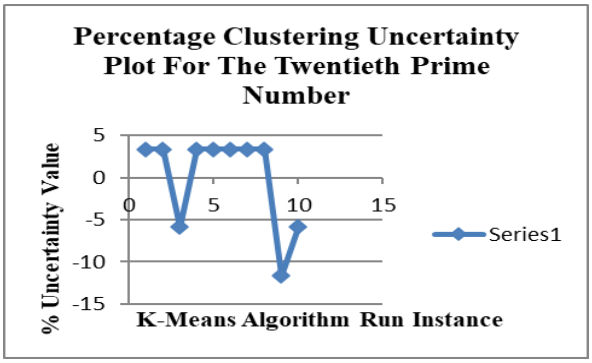

Fig 11 - Percentage Clustering Uncertainty Plot for the Twentieth Prime Number

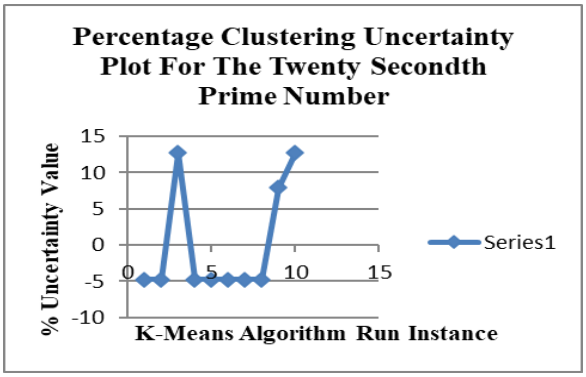

Fig 12 - Percentage Clustering Uncertainty Plot for The Twenty Secondth Prime Number

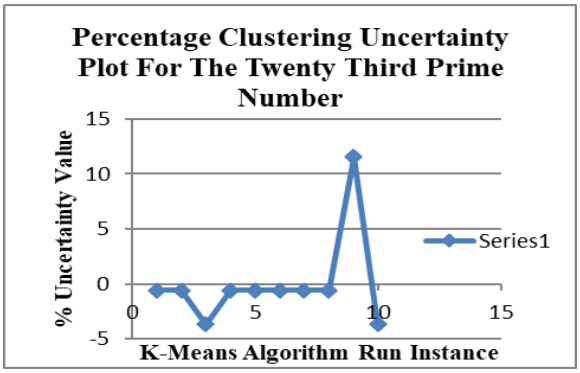

Fig 13 - Percentage Clustering Uncertainty Plot for The Twenty Third Prime Number

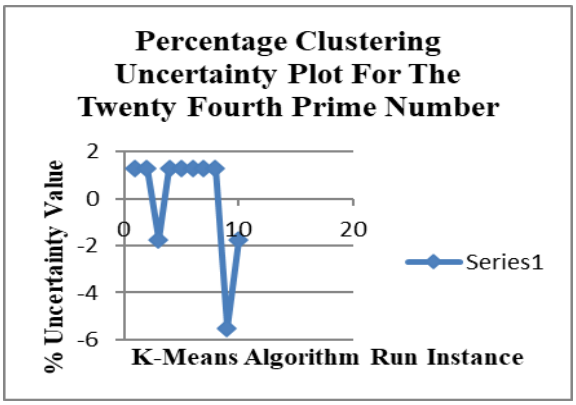

Fig 14 - Percentage Clustering Uncertainty Plot for The Twenty Fourth Prime Number

The Uncertainty Plots of the 2nd Prime through 6th Prime are same as that of the 1st Prime.
The Uncertainty Plot of the 8th Prime is same as that of the 7th Prime.
The Uncertainty Plot of the 11th Prime is same as that of the 10th Prime.
The Uncertainty Plots of the 13th Prime through 14th Prime are same as that of the 12th Prime.
The Uncertainty Plot of the 18th Prime is same as that of the 17th Prime.
The Uncertainty Plot of the 21st Prime is same as that of the 20th Prime.
The Uncertainty Plot of the 25th Prime is same as that of the 24th Prime.
Also, the Elbow Plots of all the 10 Runs of the K-Means Clustering Algorithm indicated that 5 is the Optimal Number of Clusters to be considered.

REFERENCES

[1] https://en.wikipedia.org/wiki/K-means_ clustering
[2] Stuart P. Lloyd. Least squares quantization in pcm. IEEE Transactions on Information Theory, 28(2):129– 136, 1982.
[3] J. MacQueen, Some Methods for Classification & Analysis of Multivariate Data, Fifth Berkeley Symposium, 281-297.
[4] Shengchao Du., and Lifeng Wang., Aircraft Design Optimization with Uncertainty Based on Fuzzy Clustering Analysis, ASCE, Journal of Aerospace Engineering, Vol 29 No 1 (2016)
[5] Carl Edward Rasmussen., Bernard J. de la Cruz., Zoubin Ghahramani., and David L. Wild.,

Modeling and Visualizing Uncertainty in Gene Expression Clusters Using Dirichlet Process Mixtures, IEEE/ACM Transcations On Computational Biology & Bioinformatics, Vol 6, No 4, (2009) 615

[6] Prasad I.L.N., Balaji K.V.G.D. and Kapuganti, Chitti Babu, Analysis of Uncertainty Inherent to Valuation Methodologies in Construction Industry, International Journal of Advanced Research in Engineering and Technology, 11(6), 2020, pp. 786-806. http://www.iaeme.com/IJARET/issues.asp?JType=IJARET&VType=11&IType=6

[7] file:///C:/Users/Chandini/Downloads/KARBASI-THESIS-2014.pdf

[8] https://www.researchgate.net/post/why_the_output_image_of_the_kmeans_clustering_changes_from_one_execution_to_another

[9] k-means++: The advantages of careful seeding. Arthur, D., & Vassilvitskii, S. (2007, January). In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (pp. 1027-1035). Society for Industrial and Applied Mathematics.

[10] Bachem, Olivier. et. al., Distributed and Provably Good Seedings for k-Means in Constant Rounds Proceedings of the 34 th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017

[11] Bachem, Olivier. et. al., Fast and Provably Good Seedings for k-Means, Advances in Neural Information Processing Systems 29 (NIPS 2016)

[12] Khan, Fouad. An Initial Seed Selection Algorithm for K-means Clustering of Georeferenced Data to Improve Replicability of Cluster Assignments for Mapping Application, Applied Soft Computing, Volume 12, Issue 11, November 2012, Pages 3698-3700

[13] K., Karteeka Pavan. et. al., Robust seed selection algorithm for k-means type algorithms - Optimal centroids using high density object, https://arxiv.org/ftp/arxiv/papers/1202/1202.1585.pdf