Initializing K-Means Clustering Algorithm using Statistical Information

Mohammad F. Eltibi
Islamic University of Gaza
Palestine, Gaza

Wesam M. Ashour
Islamic University of Gaza
Palestine, Gaza

ABSTRACT

K-means clustering algorithm is one of the best known algorithms used in clustering; nevertheless it has many disadvantages as it may converge to a local optimum, depending on its random initialization of prototypes. We will propose an enhancement to the initialization process of k-means, which depends on using statistical information from the data set to initialize the prototypes. We show that our algorithm gives valid clusters, and that it decreases error and time.

General Terms

Data Mining, Unsupervised Learning, Data Clustering.

Keywords

Clustering, K-means Clustering, Initial Prototypes Determination, Central Limit Theory, Normal Distribution, Maximum Likelihood Estimator.

1. INTRODUCTION

The clustering process is defined as grouping similar objects together into groups or clusters. Objects that belong to one cluster should be very similar to each other, but objects in different clusters will be dissimilar. One difficulty in this process is that we don’t have any prior knowledge about the structure of the data, or its labels, because clustering is considered to be an unsupervised learning problem [1][2]. Clustering has been considered a hot topic for decades and its applications appear in many areas, such as pattern recognition [3], data mining and knowledge discovery [4], data compression and vector quantization [5], optimization [6]. Also its applications appear in the commercial field: nowadays organizations have large volumes of data, related to their business processes, and resources, and this data can provide statistical information, so it will be useful to get some knowledge about this data to improve performance and profit. Also data clustering is useful in many non-commercial applications such as health care systems [7], marketing, monitoring systems [8], web, and etc.

Many clustering algorithms have been proposed, and there are many classifications of clustering algorithms. We focus on distance based, and density based algorithms. In distance based methods n objects are formed into k different clusters (k<n). The number of clusters to be constructed is known before hand. The best known distance based algorithm is k-means [9]; other methods such as k-medoids, PAM, CLARA and CLARANS are examples of distance based algorithms [10]. On the other hand density based methods do not require a prior knowledge of the number of clusters beforehand. Instead they require some control parameters that will be used to form clusters by analyzing the density occupied by the objects in data space. There are many density based algorithms such as DBSCAN [11] (the most famous), OPTICS [12] and DENCLUDE [13].

K-means is the best known algorithm, because it is fast and converges to acceptable results in different areas. The algorithm works as follows: the user first must define the number of clusters to be founded by k-means; k-means starts by initializing a number of prototypes equal to number of desired clusters, then makes two steps; the first is assigning each point to its closest center, then moving each prototype to the mean of its assigned points. These two steps will be repeated until it converges to a solution. K-means depends on minimizing the square sum of error (SSE) to assign each point to its cluster. Its simplicity and acceptable results mean it has a wide usage. However k-means has some drawbacks: first the user may not know in advance the number of clusters; also k-means is sensitive to the random initializing of its prototypes which may give poor clusters, since a different initialization may give different results, and this will cause k-means to converge to a suboptimal solution rather than the global optimum [2]. Another disadvantage of k-means is its sensitivity to outliers.

Our main objective is to develop enhancements to the k-means algorithm by tackling the problem of initializing the prototypes. We develop a method that finds the mean of the points using statistical information from data set, and initialize the prototypes around this mean. The rest of this paper is organized as follows: in Section 2 we will review some related works; Section 3 will discuss in detail our proposed algorithm; in Section 4 we will view the experimental results; Section 5 will conclude.

2. RELATED WORKS

The impact of initializing prototypes is significant in k-means, so there are several methods proposed to solve this problem. One of these methods was addressed by Duda and Hart [14], who proposed a recursive method for initializing the centers by running K clustering problems. Arai and Barakbah [15] proposed a hierarchical method to determine the initialization of clusters by applying k-means several times, then obtaining a set of centers from each different run; these centers will treated as a data set, and handled by a hierarchal clustering algorithm to obtain the best centers.

Lu et al [16] proposed another hierarchical initialization method to the k-means clustering problem. The core of this method is to treat the clustering problem as a weighted clustering problem so as to find better initial cluster centers based on the hierarchical approach. It depends on two major steps: first it reduces the data from the bottom up, by sampling the clusters; it then maintains clusters at the level that sampling stops, which allows them to obtain clusters centers; then based on those centers it finds the cluster prototypes of the original data by using a hierarchal method. The algorithm seems to give an acceptable result for low dimensional data sets; it suffers from the curse of the dimensionality with high dimensional data sets.

Khan and Ahmad [17] proposed an algorithm for initializing k-means prototypes based on individual attributes of the pattern, which may provide some information about initial cluster centers. The algorithm's main concept is applying k-means for each attribute to compute cluster centers for individual attributes. This is done by assuming each of attributes of the pattern space are normally distributed; they then divide the normal curve into k partitions, and apply the k-means algorithm on this attribute. They then allocate previous cluster labels to every pattern, and run k-means on the complete data set. Now each pattern has a set of class labels; a center of these classes must be found and used as prototypes for k-means. Coa et al [18] proposed a method for initializing the k-means algorithm using a neighborhood model. The cohesion degree of the neighborhood of an object and the coupling degree between neighborhoods of objects are defined based on the neighborhood-based rough set model. A new initialization method is proposed by computing cohesion for each object, and finding the one that has maximum cohesion; this will be considered as the first prototype. The next prototype will be the most coherent object satisfying maximum cohesion (after removing the selected centers). This procedure will be repeated until it finds the required number of prototypes, then for each center, it must have coupling with the samples below a threshold. If not, it will be removed and the algorithm will try to find the next center until it finds the desired number of prototypes.

3. PROPOSED ALGORITHM

The new algorithm depends on finding the best location to initialize the prototypes of k-means using statistical information from the data set. To get this information we must know the distribution of the data set. This will be done by using Central Limit Theory (CLT)[19], which states that the distribution of a sufficiently large number of independent variables, each with its own distribution, will be approximately normal. Formally the theorem can be stated as follows:

Theorem 1: Central Limit Theory

Given a data set {X1, X2, …, Xn} which contains n samples that are independent and identically distributed, each with its own distribution, then the central limit theorem asserts that for large n, the distribution of S_n is approximately normal with mean μ and variance $\frac{1}{n}\sigma^2$.

The strength of the theorem is that Sn approaches normality regardless of the shapes of the distributions of individual Xi's. We can use this notation, by considering that a data set contains several clusters, and each cluster has its own distribution that is identically distributed to that of the other clusters. Using CLT we can state that the sum of each cluster (whole data set) will have a normal distribution. Now the best initialization of prototypes is "around" the mean of this distribution. So we need to estimate the mean of data set (assumed its distribution is normal). One of the most known estimators is Maximum Likelihood Estimator (MLE) [20], it states that the desired probability distribution is the one that makes the observed data "most likely", which means that we must seek the value of the parameter vector θ that maximizes the likelihood function. The parameters of a Gaussian distribution are the mean (μ) and variance (σ). i.e. θ= {μ, σ}. Given a data set D={ X1, X2, …, Xn} , the likelihood of those objects for Gaussian distribution is:

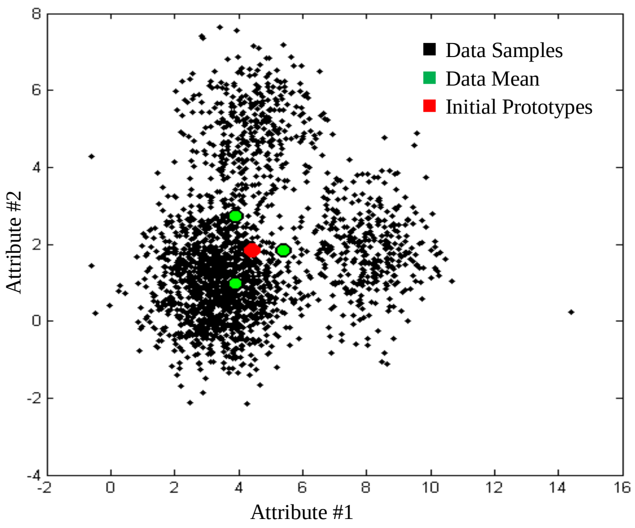

Figure 1: Data distribution (black), the mean (red), prototypes locations (green)

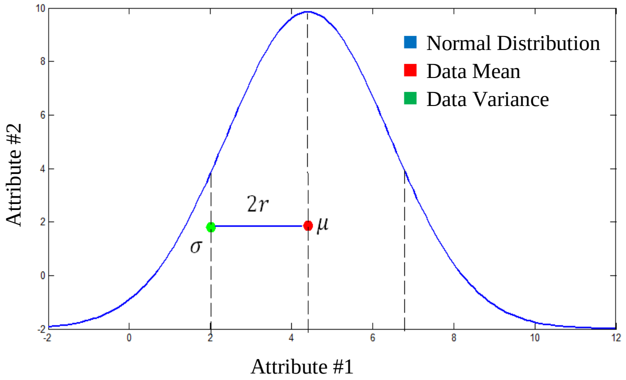

Figure 2: Calculating the radius of hypersphere using mean (red), and variance(green)

$$p(x_1, \dots, x_N|\mu, \sigma) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left\{\frac{-(x_n - \mu)^2}{2\sigma^2}\right\} \quad \dots (1)$$

and the log likelihood is

$$L(\mu, \sigma) = -\frac{1}{2} N \log(2\pi\sigma^2) - \sum_{n=1}^{N} \frac{(x_n - \mu)^2}{2\sigma^2} \quad \dots (2)$$

We can then find the values of μ and σ^2 that maximize the log likelihood by taking derivative with respect to the desired variables and solving the equation obtained. By doing so, we find that the MLE of the mean is:

$$\mu^{\hat{}} = \frac{1}{N} \sum_{n=1}^{N} x_n \quad ...(3)$$

And the MLE of the variance is

$$\sigma^{2^{\hat{}}} = \frac{1}{N} \sum_{n=1}^{N} (x_n - \mu^{\hat{}})^2 ...(4)$$

By using MLE we can estimate the parameters θ= {μ, σ }, for the normal distributed data. We use the mean μ of the samples, to be used as a location to initialize the prototypes. We cannot initialize all prototypes using the same location, this will cause all points to be assigned to one cluster, and all other clusters are considered empty. So we initiate prototypes around the mean of the data, we use a hypersphere with center equals to μ, and use its surface to initiate the prototypes, also we need to find the radius of this hypersphere to find a location for each prototype in this surface. For n-dimensional hypersphere with radius r, and $\mathrm{n}-1$ angular coordinates $\varphi_1, \varphi_2, \dots, \varphi_{n-1}$ where φ_{n-1} range over [0,360) degrees, if x_i is a point in its surface so we may compute $x_1, x_2, \dots, x_n$ from $r, \varphi_1, \varphi_2, \dots, \varphi_{n-1}$ with

$$x_1 = r \cos(\varphi_1)$$

$$x_2 = r\, sin(\varphi_1)\, cos(\varphi_2)$$

$$x_3 = r\, sin(\varphi_1)\, sin(\varphi_2)\, cos(\varphi_3)$$

$$\vdots$$

$$x_{n-1} = r\, sin(\varphi_1) \dots\ sin(\varphi_{n-2})\, cos(\varphi_{n-1})$$

$$x_n = r\, sin(\varphi_1) \dots\ sin(\varphi_{n-2})\, sin(\varphi_{n-1})$$

To make the distribution of the points on the surface uniform, we can calculate $\varphi = \frac{360}{k}$, where k is number of clusters. To imagine the process, we will describe example in 2-dimensional, we will have only one angle φ_1 , and the location is defined as

$$x_1 = r \cos(\varphi_1)$$

$$x_2 = r \sin(\varphi_1)$$

As shown in Figure 1, consider we want three clusters, so we will find the mean of data (red circle) using MLE, that will be used as a center for a circle (2-d hypersphere), then $\varphi = \frac{360}{3} = 120^{\circ}$, initializing prototypes start from angle zero, and moves on the surface by 120o, so the first point will be calculated using angle 0o, the second 120o, the last will be 240o, this is shown as green points using some value of r, this is how we calculate the new locations of the prototypes using the mean of the data. Another issue which arises here is how to calculate the best value of the radius. We here propose a method to calculate the radius, it depends on calculating the distance between the mean, and the variance, and gets half of the distance as a radius of hypersphere, As $r = \frac{dis(\mu,\sigma)}{2}$. We use Euclidean distance (any distance can be used). Figure 2 shows a normal distribution with its μ (red circle), and σ (green circle), the line between them is the distance, and we will use half of this as a value for radius r.

Initializing prototypes according to previous way, may lead to find some prototypes that are far away from samples, this is called dead prototypes problem. We define a dead prototype as a prototype that has number of assigned points below threshold ε, so a prototype is considered dead if number of assigned points is less than ε, otherwise it considered alive. To avoid having dead prototypes we will check the prototypes, if some of them are dead, we will use alive prototype, and take the farthest points from it, to be assigned to the dead one, thus make it alive, then continue the algorithm. The value of ε depends on number of samples in the data set, and may directly proportional to number of samples.

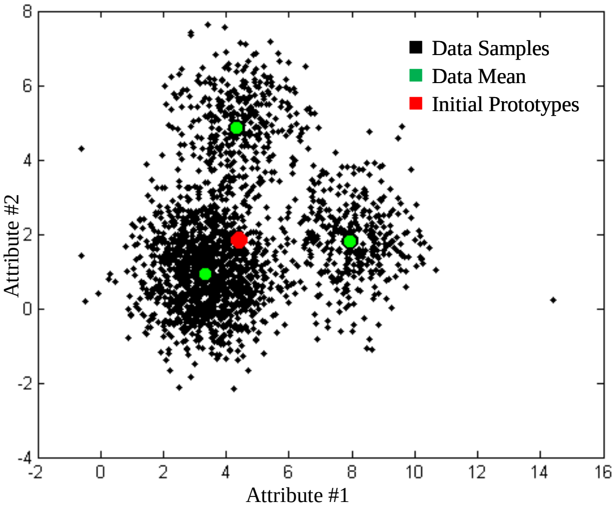

Figure3: Final result of proposed algorithm shows mean (red) of samples (black), and prototypes (green) in best location.

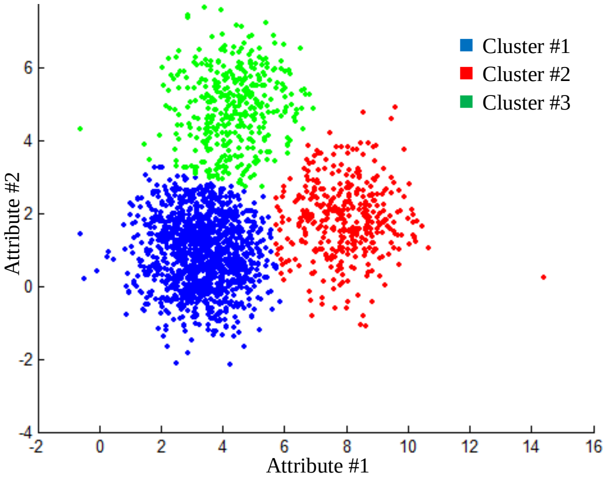

Figure 4: Final Result of proposed algorithm shows a color for each sample, where cluster1 is blue, cluster 2 is red, and cluster3 is green

Table 1: Results comparing proposed algorithm, with k-means, where samples is number of samples in data set, attributes is number of attributes in data set, error% is error rate of misclassified data, and #iteration is the number of iterations used by the algorithm.

Data Set	Samples	Attributes	Proposed Algorithm		K-means	
			Error (%)	# iteration	Error (%)	# iteration
Artificial Data Set	2,000	2	4.02%	6	8.76%	10
Iris Data Set	150	4	10.66%	10	11.33%	12
Win Data Set	178	13	24.72%	13	46.63%	13
Pima Diabetes Data Set	768	8	33.98%	16	60.03%	25

4. RESULTS AND DISCUSSION

In order to analyze the performance of our proposed method, we apply it to two kinds of experiments: artificial normal data, and real world data sets. These data (artificial and real) are labeled with the correct cluster for each observation. The error that we have calculated depends on the number of misclassified patterns, and the total number of patterns in the data set.

$$Error(\%) = \frac{Number\ of\ misclassified\ patterns}{Total\ number\ of\ patterns} \times 100$$

To compute error for k-means algorithm with random initialization of centers, we run k-means 10 times and take the average error as a performance measure. Also we take number of iteration required to converge to final solution as a measurement to the speed of the algorithm. And hence run k-means 10 times and take the average number of iterations. Following sections show the performance measurements for each data set. Also we use the Euclidian distance as a distance metric in both randomly initialization k-means, and our proposed algorithm. It also be used as a distance metric when calculating the radius of hypersphere, which will be used in our algorithm.

4.1 Artificial Data Set

This data set is a two dimensional data, made from 2,000 samples, each sample labeled with its cluster number, samples are generated from three clusters. As shown previously in Figure 1, we note how our algorithm initiate each prototype using statistical information from the data, Figure 3, and 4 show the final result of our algorithm after a small number of iterations. Our algorithm considered to be faster than original k-means (with random initialization of its centers), its speed increased about 40%, as shown in Table 1, also there is improvement in decreasing error rate.

4.2 Real Data Set

Many real data sets are used to investigate the proposed algorithm, all of them are taken from UCI [21] repository website. We take iris, wine, and Pima diabetes data sets. The iris data set is a common one, used in testing clustering algorithms, it consists of 150 samples, each belongs to one of three clusters namely Iris setosa, iris versicolor, and iris virginica, every class has 50 samples, and each sample consists of four attributes. We can note from Table 1, that our algorithm exceeds the k-means on getting low error, and has improvement on the speed, since the number of iterations used by the algorithm is decreased by 16%. The second data set (also taken from UCI) is the win data set, it consists of 768 samples, and each sample has 13 attributes, and belongs to one of three clusters. We can note that there is no improvement in the time using the new algorithm; instead we can see a good improvement in the result of the algorithm depending on the error of misclassified samples, the error decreased about 20%. The last real data set used to investigate the performance of our algorithm is the pima diabetes data set, it is a database for a female patients (samples), each has 8 attributes, every sample belongs to two classes, indicates that patient is having diabetes, or not. From this large data set we can see that our proposed algorithm has a very good performance, its performance exceeds k-means by 36%. It also decreases the error rate about 30%, this considered as good enhancement. We can say the strength of the proposed algorithm is shown when the data set is growing in size. From the previous results, we can state that our algorithm performs well and better than k-means in convergence time, especially when the data set is large. Also we can notice that the algorithm gives smaller number of errors, even when the data set is growing with its size, or with dimensions. This is naturally since it does not initialize prototypes randomly; it uses the data set to get best location to initialize these prototypes.

5. CONCLUSION

A new algorithm is proposed to solve the problems generated from randomly initialized k-means algorithm, it depends on initializing prototypes according to statistical information calculated from data, it initiates prototypes as points located on a surface of hypersphere centered in the mean of the sample. The proposed algorithm achieved good results in decreasing the time of k-means, and the error rate, its strength arising when working with very large data sets.

6. REFERENCES

[1] G. Gan, C. Ma, J Wu. "Data Clustering Theory, Algorithms, and Applications". American Statistical Association Alexandria, Virginia, 2007.

[2] P. Tan, M. Steinbach, V. Kumar. “Introduction to Data Mining”. Addison-Wesley , 2006.

[3] D. Fisher. “Knowledge acquisition via incremental conceptual clustering”. Machine Learning, 1987, pp. 39–172.

[4] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy. "Advances in Knowledge Discovery and Data Mining". AAAI Press, 1996.

[5] A. Gersho, R.M. Gray. "Vector Quantization and Signal Compression". KAP, 1992.

[6] P.S. Bradley, O.L. Mangasarian, W.N. Street. "Clustering via concave minimization". Advances in Neural Information Processing System, MIT Press, vol. 9, 1997, pp. 368–374

[7] J. Aguilar. "Resolution of the Clustering Problem using, Genetic Algorithms". International Journal of computers, vol. 1, 2007.

[8]R. Vaarandi, "A Data Clustering Algorithm for Mining Patterns from Event Logs", Proceedings of the 2003 IEEE Workshop on IP Operations and Management. IEEE. 2003.

[9] Q.J. Mac. "Some methods for classification and analysis of multivariate observations". In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967, pp. 281-297.

[10] R. T Ng, J. Han. "Efficient and Effective Clustering Methods for Spatial Data Mining", Proceedings of 20th International Conference on Very Large Databases. Santiago de Chile, 1994, pp. 144 – 155.

[11] E. Martin, H. Kriegel, J. Sander, X. Xu. "A Density Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Proceedings of second International Conference on Knowledge Discovery and Data Mining, Kluwer Academic Publishers, 1996, pp. 169- 194.

[12] M. Ankerst, M. M. Breunig, H. Kriegel, J. Sander. "OPTICS: Ordering Points to Identify the Clustering Structure". Proceedings of ACM SIGMOD. Pergamon Press, 1999, pp. 5761 -5767.

[13] A. Hinneburg, H. Gabriel. "An Efficient Approach to Clustering in Large Multimedia Databases with Noise", Proceedings of Knowledge Discovery and Data Mining. AAAI Press, 1998, pp. 58 -65.

[14] R.O. Duda, P.E. Hart. "Pattern Classification and Scene analysis". John Wiley and Sons, NY. 1973.

[15] K. Arai, A. R. Barakbah. "Hierarchical K-means: an algorithm for centroids initialization for K-means". Reports of the Faculty of Science and Engineering. Saga University, vol. 36, No.1, 2007, pp. 25-31.

[16] J. F. Lu, J. B. Tang, Z. M. Tang, J.Y. Yang. "Hierarchical initialization approach for K-Means clustering". Pattern Recognition Letters, vol. 29, April 2008, pp. 787-795.

[17] S. Khan, A. Ahmad. "Cluster center initialization algorithm for K-means clustering". Pattern Recognition Letters, vol. 25, August 2004, pp. 1293-1302.

[18] F. Caoa, J. Liang , G. Jiang . "An initialization method for the k-Means algorithm using neighborhood model". Computers & Mathematics with Applications, vol. 58, August 2009, pp. 474-483.

[19] R. M. Dudley. "Uniform Central Limit Theorems". Cambridge University Press, 2008.

[20] I. Myung. "Tutorial on maximum likelihood estimation". Journal of Mathematical Psychology, vol 47, 2003.

[21] UCI Repository [Online]. Available: http://archive.ics.uci.edu.