Journal of
Computer Science & Systems Biology

Jaya RamaKrishnaiah et al., J Comput Sci Syst Biol 2012, 5:2
DOI: 10.4172/jcsb.1000091

Research Article Open Access

Entropy Based Mean Clustering: An Enhanced Clustering Approach

V.V. Jaya RamaKrishnaiah[1]*, K. Ramchand H Rao[2] and R. Satya Prasad[3]

[1]A.S.N. Degree College, Tenali, AP, India
[2]A.S.N. Women's Engineering College, Nelapadu, AP, India
[3]Acharya Nagarjuna University, Guntur, AP, India

Abstract

Many applications of clustering require the use of normalized data, such as text data or mass spectra mining data. The K –Means Clustering Algorithm is one of the most widely used clustering algorithm which works on greedy approach. Major problems with the traditional K mean clustering is generation of empty clusters and more computations required to make the group of clusters. To overcome this problem we proposed an Algorithm namely Entropy Based Means Clustering Algorithm. The proposed Algorithm produces normalized cluster centers, hence highly useful for text data or massive data. The proposed algorithm shows better performance when compared with traditional K Mean Clustering Algorithm in mining data in terms of reducing time, seed predications and avoiding Empty Clusters.

Keywords: K-Mean; Entropy; Euclidian distance; Clustering

Introduction

While data collection methodologies have become increasingly sophisticated in recent years, the problem of inaccurate data continues to be a challenge for many data mining problems. This is because data collection methodologies are often inaccurate and are based on incomplete or inaccurate information. For example, the information collected from surveys is highly incomplete and either needs to be imputed or ignored altogether. In other cases, the base data for the data mining process may itself be only estimation from other underlying phenomena. In many cases, a quantitative estimation of the noise in different fields is available. An example is illustrated in [1], in which error driven methods are used to improve the quality of retail sales merchandising. Many scientific methods for data collection are known to have error-estimation methodologies built into the data collection and feature extraction process.

Exploratory data analysis processes often make use of clustering techniques. This can be used to look for groups of similar objects according to some metrics. Properties can be considered as well. Many methods can provide relevant partitions on one dimension (say objects or properties) but they suffer from the lack of explicit cluster characterization, i.e., what are the properties that are shared by the objects of a same cluster.

Clustering is one of the most important research areas in the field of data mining. Clustering means creating groups of objects based on their features in such a way that the objects belonging to the same groups are similar and those belonging to different groups are dissimilar. Clustering is an unsupervised learning technique. The main advantage of clustering is that interesting patterns and structures can be found directly from very large data sets with little or none of the background knowledge. Clustering algorithms can be applied in many domains.

K-means clustering is a method that partitions n data points within a vector space into k distinct clusters. Points are allocated to the closest cluster and cluster locations arise naturally to fit the available data. K-means minimizes intra-cluster variance, that is, clusters form that minimize the sum of the squared distances between data points and the center (centroid) of their containing cluster. However, k-means is not guaranteed to find a global minimum.

The k-means algorithm [2,3] is successful in producing clusters for many practical applications. But the computational complexity of the original k means algorithm is very high, especially for large data sets. Moreover, this algorithm results in different types of clusters depending on the random choice of initial centroids [4]. Many difficulties in comparing quality of the clusters produced, for example for different initial partitions of values of k affect outcome, does not work well with non-globular clusters. Several attempts were made by researchers for improving the performance of the k-means clustering algorithm.

In this paper, we have presented an enhanced approach, which eliminates the unnecessary computations in making the partition of the data. Here the basic execution of the k-means algorithm is preserved along with all its necessary characteristics. With the proposed algorithm, the complexity of the mechanism was reduced by adopting the entropy of the seed of the cluster. The term entropy defines the number of same instances of the dataset.

Problem statement

To compare the two algorithms using normal distribution data points. This investigate can be used two unsupervised clustering methods, namely K-Means, Entropy based k-means are examined to analyze based on the distance between the input data points. The clusters are formed according to the distance between data points and cluster centers are formed for each cluster. For implementation

***Corresponding author:** V.V Jaya RamaKrishnaiah, A.S.N. Degree College, Tenali, AP, India, E-mail: jkvemula@yahoo.com

Received May 08, 2012; **Accepted** May 25, 2012; **Published** June 07, 2012

Citation: Jaya RamaKrishnaiah VV, Ramchand H Rao K, Satya Prasad R (2012) Entropy Based Mean Clustering: An Enhanced Clustering Approach. J Comput Sci Syst Biol 5: 062-067. doi:10.4172/jcsb.1000091

plan, we take the datasets from UCI Machine Learning Repository. The implementation work was used in Advanced Java, MS-Excel and MATLAB software. The execution time is calculated in milliseconds. This paper deals with a method for improving efficiency of the k-means algorithm and analyze the elapsed time is taken, predicting best seed points and removing the empty clusters by entropy based k-means is less than k means algorithm.

Factors Drives Towards, Proposed Work

This segment describes the original k-means clustering algorithm. The idea is to classify a given set of data into k number of transfer clusters, where the value of k is fixed in advance. The algorithm consists of two separate phases: the first stage is to define k centroid, one for each cluster [5]. The next stage is to take each point belonging to the given data set and associate it to the nearest centroid.

The k-means algorithm is an evolutionary algorithm that gains its name from its method of operation. The algorithm clusters observations into k groups, where k is provided as an input parameter. It then assigns each observation to clusters based upon the observation's proximity to the mean of the cluster.

The cluster's mean is then recomputed and the process begins again. Here's how the algorithm works:

1. The algorithm arbitrarily selects k points as the initial cluster centers ("means").
2. Each point in the dataset is assigned to the closed cluster, based upon the Euclidean distance between each point and each cluster center.
3. Each cluster center is recomputed as the average of the points in that cluster.
4. Steps 2 and 3 repeat until the clusters converge. Convergence may be defined differently depending upon the implementation, but it normally means that either no observations change clusters when steps 2 and 3 are repeated or that the changes do not make a material difference in the definition of the clusters.

In K-Mean Algorithm, a set D of N patterns $\{x_1,x_2,\ldots..,x_n\}$ of dimensions d is partitioned into K clusters denoted by $\{C_1,C_2,C_3,\ldots\ldots C_n\}$ with the objective function J

$$J=\sum_{j=1}^{k}\sum_{i=1}^{n}\left|x_i^{(j)}-C_j\right|^2$$

where $||x_i^{(j)}-C_j||^2$ is a choosen distance measure between a data point $x_i^{(j)}$ and the cluster centre C_j, is an indicator of the distance of the n data points from their respective cluster centre's.

This K mean algorithm is good and produces qualitative results, it is struggles with the more number of computations, not put good effects on the non-global clusters, and other major problem is handling the null clusters.

The Entropy Based Means Clustering Algorithm

The proposed Entropy Based Means Clustering algorithm, reduces the significant limitations observed in the basic K-mean clustering technique The Entropy based Mean algorithm is slightly modifies K Mean Clustering method. In this algorithm can be used more effective than normal k-means algorithm. The proposed algorithm works in the three phases. In the first phase it computes the minimum points of the each seed (element or item) in the data set and then arranges the seed elements in the order of their seed entropy (For example (Seed-Entropy): 1-10,2-5,3-9,4-6,5-1, then it arranges the data as 1,3,4,2,5 .i.e. data arranged descending order of the entropy). In the second phase, it makes the candidate set, this candidate set is unique in nature, i.e. it does not consisting of duplicated elements. In the third phase the clustering was applied on the Euclidian distances, and remaining elements, which were not in candidate sets were placed in according to the native elements, were resided.

Arranging the data in the descending order of the entropy

This Phase identifies the entropy of each seed in the data set $D=\{x_1, x_2, x_3\ldots x_n\}$ and arranges them in the descending order of the seed entropy. Here entropy was calculated as the number of elements of the same kind.

For example D={1,2,1,2,4,4,4,1,4,1,1,3} and the entropies of each were as follows

Seed	1	2	3	4
Entropy	5	2	4	1

Then the data is rearranged as 1:5, 3:4, 2:2, 4:1.

Identification of candidate set C from D: This phase determines the number of candidate seeds in the dataset. Here in our sample we have 4 data seeds, instead of 12 data seeds. By this we can reduce the number of computations, for making a cluster. C= {1,3,2,4}

Making the clusters by using defined K and C: This phase, works in three steps as follows

1. The clustering technique was applied based on the Euclidian distances by using Candidate Set
2. Rearranging the elements remaining elements in D.
3. This is optional step, which handles the empty clusters, if any. In this step we can remove the clusters, which does not consisting of any seed. For the Entropy based, there is no possibility for forming the empty clusters.

For our example, if k=2, then cluster C1={1,1,1,1,1,2,2,}, C2={3,3,3,3,4}.

K=3, then cluster C1={1,1,1,1,1}, C2={3,3,3,3}, C3={2,2,4}.

Algorithm

Input: D = {d1, d2... dn} // set of *n* data items

k // Number of desired clusters

Output: A set of *k* clusters.

Procedure

1. Fetch the each data element in the D and estimate the entropy of each data element.

2. Sort the data elements in descending order of entropies and call them as seeds.
3. Make the Candidate data set C such that no duplicates seed in C. and make one duplicate candidate set DC.
4. a) Set mean for each cluster CL_k as 0 and call it as Cluster Centre CC.

   b) Assign a seed to every cluster CL_k from the candidate set C.
5. Recompute the mean of each CL.
6. For each seed-point C*i* remain in C, find the closest centroid CC_j and assign C_i to cluster *j*.
7. a) Place the seed point C_i to the cluster CC_k such that the seed point distance is closer to the present nearest Distance.

   b) Detach C_i from C

   c) Repeat the step 5.
8. Repeat Step 6 to 7, until Candidate Set C becomes empty and convergence was made.
9. For each element in {D-DC} do the following step

   a) Compare each CL_K seeds with the data seeds in {D-DC}.

   b) Place the seeds in {D-DC} into the corresponding CL_k.

The above algorithm reveals that the new clustering scheme is exactly similar to the original k-means algorithm process, except some differences like making of Candidate set C preparation, which reduces the number of computations and rearrangement of data seeds.

Rate of convergence of the entropy based clustering algorithm

In the proposed algorithm, an iteration starts with a set of ole center $CC_k^{(old)}$, the data elements were distributed among the clusters depending upon the minimum Euclidean distance, and then set of new clusters $CC_k^{(new)}$ is generated by averaging the data elements.

This center updation can be mathematically described as follows:

$$CC_k^{(new)} = \frac{1}{n_k}\left(\sum_{x_j \in CL_k} x_j\right)$$

where n_k is the number of elements in cluster CL_k, If new centers $CC_k^{(new)}$ do not match exactly with the old center $CC_k^{(old)}$, the algorithm does the next iteration assuming $CC_k^{(new)}$ as $CC_k^{(old)}$.

Let CL be the cluster consisting of the elements represented by $CL = \{x_1, x_2, x_3, x_4, \ldots\ldots x_n\}$ and the corresponding cluster center $CC^{(t)} = x$, where $x \neq \frac{x_1 + x_2 + \ldots\ldots + x_n}{m}$. The subsequent centers can be obtained as follows:

$$CC^{(t+1)} = \frac{\sum_{i=1}^{m} x_i}{(m+1)} + \frac{x}{(m+1)}$$

$$CC^{(t+2)} = \frac{\sum_{i=1}^{m} x_i}{(m+1)} + \frac{\sum_{i=1}^{m} x_i}{(m+1)^2} + \frac{x}{(m+1)^2}$$

Similarly for the r[th] iteration,

$$CC^{(t+r)} = \frac{\sum_{i=1}^{m} x_i}{(m+1)} + \frac{\sum_{i=1}^{m} x_i}{(m+1)^2} + \frac{\sum_{i=1}^{m} x_i}{(m+1)^3} + \ldots + \frac{\sum_{i=1}^{m} x_i}{(m+1)^r} + \frac{x}{(m+1)^r}$$

$$= \frac{\sum_{i=1}^{m} x_i}{(m+1)} X \frac{1-\left(\frac{1}{m+1}\right)^r}{1-\frac{1}{m+1}} + \frac{x}{(m+1)^r}$$

All the above equations are in G.P., so combine all the equations and we get the converged condition for entropy based mean clustering as follows:

$$\frac{CC^{t+r}}{n} = \frac{\sum_{i=1}^{m} x_i}{m^r}$$

And the converged center for all clusters, say $CL_1 = \{x_1, x_2, x_3, x_4, \ldots\ldots x_p\}$ and $CL_2 = \{y_1, y_2, y_3, \ldots\ldots y_q\}$ the overall process can be defined as

$$CC_i^{(i)} = \frac{x_1 + x_2 + x_3 + \ldots. + x_p + y_1 + y_2 + y_3 + \ldots.. + y_q + CC_i^{(0)}}{(p+q+1)}$$

Satisfying the two conditions

$$CC_i^{(1)} = CC_i^{(2)}$$

$$CC_i^{(1)} \neq CC_{i+1}^{(2)}$$

with the above equations, we can reduce the empty clusters.

Illustration

This section provides the performance comparison of the conventional K-Mean Clustering and Entropy Based Mean Clustering in terms of handling null clusters, Seed Predictions and Time Complexity of the clusters.

To make this testing, we used the dataset extracted from UCI Machine Learning Dataset [6] called "abalone" dataset with 50 instances and 9 attributes. To make this experiment, we use the attribute called "Rings". Table 1 describes the dataset used to experiment the K-mean and Entropy based clustering mechanisms.

Handling of empty clusters: Here, we shall experimentally proved, how the Entropy based clustering algorithm overcomes the problem with K-mean clustering, in the view of avoiding empty clusters. Table 2 illustrates about then No. of samples or elements allocated to each cluster.

From the Table 2, we notice, K-mean algorithm was struggling with empty clusters at cluster 10,15. The entropy based clustering does not affected by empty clusters. And another point we notices is accuracy in each cluster is good as compared with K-Means.

Gender	Length	Diameter	Height	Whole Height	Shucked Weight	Viscera Weight	Shell Weight	Rings
M	0.46	0.37	0.10	0.51	0.22	0.10	0.15	15
M	0.35	0.27	0.09	0.23	0.10	0.05	0.07	7
F	0.53	0.42	0.14	0.68	0.26	0.14	0.21	9
M	0.44	0.37	0.13	0.52	0.22	0.11	0.16	10
I	0.33	0.26	0.08	0.21	0.09	0.04	0.06	7
I	0.43	0.30	0.10	0.35	0.14	0.08	0.12	8
F	0.53	0.42	0.15	0.78	0.24	0.14	0.33	20
F	0.55	0.43	0.13	0.77	0.29	0.15	0.26	16
M	0.48	0.37	0.13	0.51	0.22	0.11	0.17	9
F	0.55	0.44	0.15	0.89	0.31	0.15	0.32	19
F	0.53	0.38	0.14	0.61	0.19	0.15	0.21	14
M	0.43	0.35	0.11	0.41	0.17	0.08	0.14	10
M	0.49	0.38	0.14	0.54	0.22	0.10	0.19	11
F	0.54	0.41	0.15	0.68	0.27	0.17	0.21	10
F	0.47	0.36	0.10	0.48	0.17	0.08	0.19	10
M	0.50	0.40	0.13	0.66	0.26	0.13	0.24	12
I	0.36	0.28	0.09	0.29	0.10	0.04	0.12	7
F	0.44	0.34	0.10	0.45	0.19	0.09	0.13	10
M	0.37	0.30	0.08	0.26	0.10	0.04	0.10	7
M	0.45	0.32	0.10	0.38	0.17	0.08	0.12	9
M	0.36	0.28	0.10	0.25	0.10	0.06	0.08	11
I	0.38	0.28	0.10	0.23	0.08	0.05	0.09	10
F	0.57	0.44	0.16	0.94	0.43	0.21	0.27	12
F	0.55	0.42	0.14	0.76	0.32	0.21	0.20	9
F	0.62	0.48	0.17	1.16	0.51	0.30	0.31	10
F	0.56	0.44	0.14	0.93	0.38	0.19	0.30	11
F	0.58	0.45	0.19	1.00	0.39	0.27	0.29	11
M	0.59	0.45	0.14	0.93	0.36	0.23	0.28	12
M	0.61	0.48	0.18	0.94	0.39	0.22	0.30	15
M	0.58	0.43	0.14	0.86	0.39	0.23	0.20	11
M	0.58	0.47	0.17	1.00	0.39	0.24	0.33	10
F	0.68	0.56	0.17	1.64	0.61	0.28	0.46	15
M	0.67	0.53	0.17	1.34	0.55	0.36	0.35	18
F	0.68	0.55	0.18	1.80	0.82	0.39	0.46	19
F	0.71	0.55	0.20	1.71	0.63	0.41	0.49	13
M	0.47	0.36	0.11	0.48	0.23	0.12	0.13	8
F	0.54	0.48	0.16	1.22	0.53	0.31	0.34	16
F	0.45	0.36	0.11	0.52	0.24	0.12	0.15	8
F	0.58	0.45	0.14	0.88	0.38	0.20	0.26	11
M	0.36	0.29	0.09	0.33	0.13	0.09	0.09	9
F	0.45	0.34	0.11	0.43	0.19	0.09	0.12	9
F	0.55	0.43	0.14	0.85	0.36	0.20	0.27	14
I	0.24	0.18	0.05	0.07	0.03	0.02	0.02	5

I	0.21	0.15	0.06	0.04	0.03	0.02	0.01	5
I	0.21	0.15	0.05	0.04	0.02	0.01	0.02	4
I	0.39	0.30	0.10	0.20	0.09	0.05	0.08	7
M	0.47	0.37	0.12	0.58	0.29	0.23	0.14	9
F	0.46	0.38	0.12	0.46	0.18	0.11	0.15	7
I	0.33	0.25	0.07	0.16	0.08	0.03	0.05	6
F	0.53	0.43	0.16	0.84	0.35	0.21	0.25	9

Table 1: Abalone Database.

Mechanism	No.of Instances	C=2 (No of samples)	C=5 (No of samples)	C=10 (No of samples)	C=15 (No of samples)	Empty Clusters
K –Mean Clustering	50	C1=38, C2=12	C1=4,C2=13, C3=6,C4=16, C5=11	C1=6,C2=4, C3=4,C4=8, C5=3,C6=6, C7=4,C8=9, C9=0,C10=6	C1=1,C2=4,C3=3, C4=1,C5=3,C6=8, C7=4,C8=1,C9=6, C10=2,C11=6,C12=8, C13=0,C14=3,C15=0	Yes, at Clusters number C=10, C=15
EBM Clustering	50	C1=29, C2=21	C1=8,C2=11, C3=10,C4=10, C5=11	C1=8,C2=8, C3=6,C4=10, C5=3,C6=4, C7=3,C8=4, C9=2,C10=2	C1=8,C2=8,C3=6, C7=3,C8=2,C9=2, C10=2,C11=3,C12=2, C13=1,C14=1,C15=1	No

Table 2: Description of Data in Each cluster.

Mechansm	C=2	C=5	C=10	C=15
K-Mean Clustering	C1=8.94, C2=16.27	C1=4.00, C2=6.61 C3=15.00, C4=9.50 C5=11.50	C1=11.00, C2=5.00, C3=19.00,C4=10.00, C5=8.00,C6=15.00, C7=12.25,C8=9.00, C9=0,C10=7.00	C1=13.00, C2=19.00,C3=8.00, C4=6.00,C5=12.00, C6=10.00,C7=15.50 C8=4.00,C9=7.00, C10=14.00, C11=11.00, C12=9.00,C13=0, C14=5.00,C15=0
EBM Clustering	C1=12.96, C2=7.52	C1=10.00, C2=8.72, C3=11.5, C4=6.2, C5=16.45	C1=10.00,C2=9.00, C3=11.00,C4=6.20, C5=15.00,C6=12.50, C7=8.00,C8=16.00, C9=16.00, C10=14.00	C1=10.00,C2=9.00, C3=11.00,C4=7.00, C5=15.00,C6=12.00, C7=8.00,C8=19.00 C9=16.00,C10=14.00, C11=4.25,C12=20.00, C13=18.00,C14=13.00, C15=6.00

Table 3: Centers of Each Cluster.

The seed predictions of the clusters: Table 3 illustrates the centers of the cluster (C – number). From the Table 3, we notice, average size of the clusters in EBM Clustering is almost uniform, in the every case of cluster no: 2, 5, 10, and 15. But in the case of K-mean the cluster average cluster size is not uniform. This may effects the size and shape of the cluster.

Table 4, illustrates about the run time (ms) differences between the K-Mean and EBM Clustering Methods for size of the dataset is 50 instances. Figure 1, shows the comparative runtime differences between the algorithms. From the Figure 1, we notice at runtimes for EBM clustering almost all same for all clusters, but in the case of K-mean the run time are increases as the number of clusters is increases. Figure 2 final clusters obtained from the EBM Clustering Technique.

Method	C=2	C=5	C=10	C=15
K Mean Clustering	16	18	22	25
EBM Clustering	14	15	16	16

Table 4: Run time(ms).

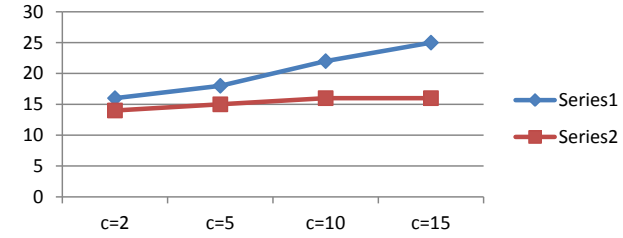

Figure 1: Comparative runtimes between K-Mean Clustering and EBM Clustering.

Conclusion

The performance of K-Mean and Entropy Based Mean Clustering Algorithm are evaluated for the dataset shown in Table 1. The results shown in Table 2 indicates that K-Mean Clustering Algorithm formed two Empty Clusters, when the number of clusters for the dataset are equal to 10,15, where as Entropy Based Mean Clustering Algorithm doesn't formed any Empty Clusters. Similarly Table 3 shows Entropy

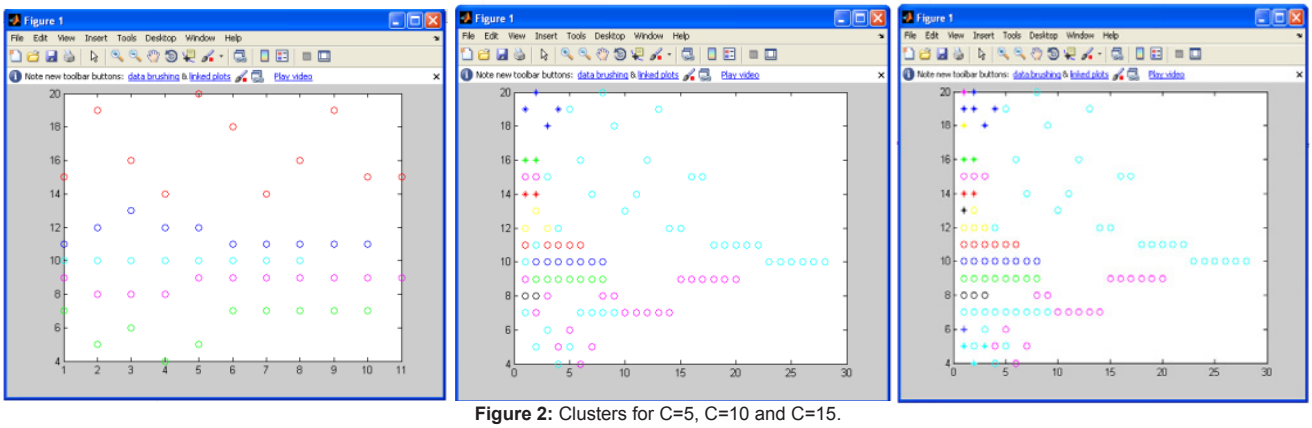

Figure 2: Clusters for C=5, C=10 and C=15.

Based Mean Clustering gives uniform Seed predications compared with K-Mean Clustering. And also Table 4 and Figure 1 explain that the execution time for taken dataset is less than that of K- Mean Clustering Algorithm. Hence we conclude Entropy Based Mean Clustering is preferable algorithm for the given conditions.

References

1. Konstantinos G, Derpanis (2005) Mean Shift Clustering.
2. Velmurugan T, Santhanam T (2010) Computational complexity between k means and k-medoid clustering algorithm for normal and uniform distribution of data points. Journal of Computer Science 6: 363-368.
3. Margaret HD (2006) Data Mining- Introductory and Advanced Topics. Pearson Education India.
4. Pang-Ning T, Steinback M, Vipin K (2007) Introduction to Data Mining. Pearson Education India.
5. Rafail O, Yuval R, Schulman LJ, Chaitanya S (2006) The Effectiveness of Lloyd-type Methods for the k-Means Problem. 165-176.
6. http://archive.ics.uci.edu/ml/datasets/Abalone