International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 6 Issue: 5 89 - 93

Generalized Markov Chain Monte Carlo Initialization for Clustering Gaussian Mixtures Using K-means

Ritu Rajawat
CSE Dept, RN Modi College of Engineering
Kota, India
riturajawat875@gmail.com

Iti Sharma
CSE Dept, RN Modi College of Engineering
Kota, India
itisharma.uce@gmail.com

Abstract— Gaussian mixtures are considered to be a good estimate of real life data. Any clustering algorithm that can efficiently cluster such mixtures is expected to work well in practical applications dealing with real life data. K-means is popular for such applications given its ease of implementation and scalability; yet it suffers from the plague of poor seeding. Moreover, if the Gaussian mixture has overlapping clusters, k-means is not able to separate them if initial conditions are not good. Kmeans++ is a good seeding method with high time complexity. It can be made fast by using Markov chain Monte Carlo sampling. This paper proposes a method that improves seed quality and retains speed of sampling technique. The desired effects are demonstrated on several Gaussian mixtures.

Keywords—*clustering, k-means, initializing k-means, kmeans++, markov chains*

*****

I. INTRODUCTION

Clustering is an unsupervised machine learning task that requires a set of objects to be partitioned into non-overlapping subsets based on the similarity relations among objects. The similarity relations are computed based on the type of data and generally for numeric data Euclidean distance or other geometric distance is used. Often clustering is presented as an optimization problem that attempts to minimize the differences or distances among the members of same cluster and maximize the distances between different clusters. Such optimization objective function is termed as squared sum of errors when a standard k-means algorithm is used for clustering. Kmeans [1] is a popular clustering method due to the flexibility and ease of implementation. It can be well adapted to variety of applications. Also, it scales well as it is linear in size of data. In case when Gaussian mixtures are to be clustered, kmeans is most preferable due to the geometric distance based objective function. But the only shortfall is that quality of cluster output may be affected if inherent cluster structure consists of overlapping clusters. Separation of clusters in the output through kmeans depends on its initial conditions. Initialization of kmeans is well researched problem and remains a challenge till date. The time versus quality trade-off is prominent. In order to have very accurate estimates of cluster centres before the algorithm actually commences requires much time. If these initial estimates of centres are very bad, the actual output is also of poor quality.

Kmeans is an iterative algorithm having two main steps per iteration. A dataset of n point of m dimensions is to be divided into k groups or clusters. Each cluster is represented by a positional representative called centroid. The algorithm takes input initial positions (values) of the k centroids. During each iteration, every point is assigned a cluster label of its nearest centroid. After one such pass through entire data, centroids are updates as mean values of the points assigned their labels. These steps of cluster assignment and centroid update are repeated until the centroids get stable. The initial values of centroids play a significant role in deciding the output cluster structure. Hence, many research works have focused on this problem.

K-means was suggested originally by Lloyd [1] and Forgy [2] independently. The method is still the standard version of k-means. They suggested random initialization for centroids. Bradley and Fayyad [3] suggested subsampling of dataset and then pick the centroids that produce best cluster structure in terms of objective function value. But solely values of objective function cannot determine the quality of output if data consists of Gaussian mixtures. Hence, even though it consumes time, the subsampling technique may fail in certain situations. Some linear time methods suggested by Agha [4], are based on uniform distribution of values in the object space and do not consider the density of objects within the subregions of object space. Method by Al-Daoud and Roberts [5] considers dividing the object space into subregions and takes into account the distribution of objects within these sub-hypercubes. Yet, due to time complexity and complicated computations this method did not gain popularity. Simple theory of statistics can be applied to observe principal components within the data and decide centroids accordingly. Su and Dy [6] suggest such techniques. Certain distance based approaches are KKZ [7], Maximin [8] and kmeans++ [9]. Kmeans++ includes theory of statistics and probability into the maximin technique. By far it has become most popular after the random technique to initialize kmeans. A comprehensive study of initialization methods is [10].

The only drawback is kmeans++ involves too many distance computations and is very slow. Bachem et al [11] have attempted to estimate the centroids through kmeans++ style of picking each successive centroid to be furthest point from already selected centroids and use Markov Chain Monte Carlo sampling therein. This eliminates the need of many distance computations and accelerates the initialization process many times over. This heavy gain in time comes at cost of slight fall in quality. This paper attempts to improve the quality of afkmc2 [12] method without increasing the runtime in proportion. Through experiments we derive a generalized version through which a user can control the time cost and quality of centroid estimate through a single control parameter.

II. Background and Related Work

Bachem et al [11] have recently suggested how Markov Chain Monte Carlo sampling can be coupled with the kmeans++ method to achieve a very fast initialization method. To discuss these schemes, consider a formal setting where X denotes a dataset of n points of m dimensions, that is any point $x \in X$ implies $x \in \mathbb{R}^m$. Let there be a finite set $C \subset \mathbb{R}^m$. Distance of any point x from C is defined as $d(x,C)^2 = \min_{c \in C} \|x - c\|_2^2$, where $\|\,.\,\|_2^2$ denotes the squared Euclidean distance. In the process of selecting a set of initial centres, the points are picked one by one and any instant C denotes the set of centres selected till then. The basic idea to which many researchers agree is that any successive centre should be the point that has maximum distance from already selected centres in order to get a full coverage of the object space. If instead of deterministic selection as in maximin method, a probabilistic approach is used then it requires some kind of sampling based on these distances from centres.

D^2– sampling : Given a set of centres C, this sampling strategy samples a point proportional to the probability based on distance of the point from centres, that is $p(x|C) = \frac{d(x,C)^2}{\sum_{x' \in X} d(x',C)^2}$. This strategy involves squared distances, hence the name. It is used in kmeans++. The first centre is uniformly sampled from X. rest all k-1 centres are sampled using probability given in (2). As is seen in (1), if C changes, the distances need to be recomputed. Thus, this method is of computation cost $\Theta(nkm)$.

MCMC sampling : Bachem et al [11] propose to sample successive centres by approximations derived from Markov chains. The D^2 sampling is an exact sampling method, and if approximated properly, can be sped up. The first seed is selected uniformly from X. Thereafter, other centres are picked by constructing a Markov chain of length r using the Metropolis – Hastings algorithm with an independent and uniform distribution $1/n$. At every iteration of selecting a centre, distances between only these r points and C need to be computed and the probability used for selection is $\min\left(1, \frac{d(y_j,C)^2}{d(c_{j-1},C)^2}\right)$, where y_j is the candidate at iteration j, $j \in [2,3,\dots,k]$. This technique assumes that the Markov chain is constructed of a length bounded by $\Theta(k \log^2 n \log k)$. If this assumption is not held, the quality of solution is not guaranteed. But verifying whether this assumption holds or not is too much time consuming and may be more expensive than the algorithm itself.

AFKMC2 : To free the above method from assumption over the distribution, Bachem et al [12] proposed a different distribution. The trade off is increasing chain length in the Markov chain process and increasing the quality of solution. The first centre is uniformly sampled. Then all successive centres are sampled according to a distribution that combines the true D2 sampling and a regularizing factor. Their proposal is following distribution: $p(x|c_1) = \frac{1}{2}\frac{d(x,c_1)^2}{\sum_{x' \in X} d(x',c_1)^2} + \frac{1}{2}\frac{1}{|X|}$, where, c_1 is the first centre. Thus, the sampling is done with respect to the first centre.

III. Proposed Method

This paper proposes an initialization technique for k-means method with a view to retain the effectiveness of k-means++ and alleviate its drawback of high time complexity. The sampling idea of AFKMC2 is indeed a good approach to speed up the initialization that trades off the solution quality of kmeans++ but follows its style. We emphasize here that any of the method that relies on D2 sampling has a common step – its initial centre is always randomly selected from the dataset. To highlight the possible cases when such a random initial centre may render entire method to be a poor initialization, consider following discussion.

In case, the first centre gets chosen such that it is very close to the mean value of entire dataset. In this case, several points will have approximately equal probability of being chosen as second centroid. And second centroid will always lie on the "edge" of the cluster structure. Consecutively, the successive centres picked would be from other "edges" of the object space. Fig1 illustrates this case. This would mean very slow convergence. Fig 2 shows the same effect in Touching dataset [13] where a wrong choice of first centroid would lead to clusters very different from ground truth. But this will not be that poor if clusters are well separated as in Fig 3, where the three clusters are well defined and the three centroids are good initialization.

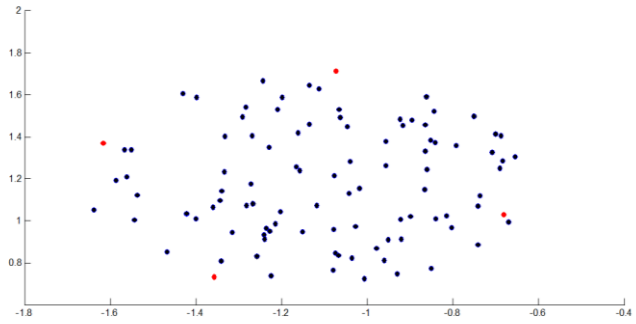

Fig 1. Object Space with too much overlap in cluster

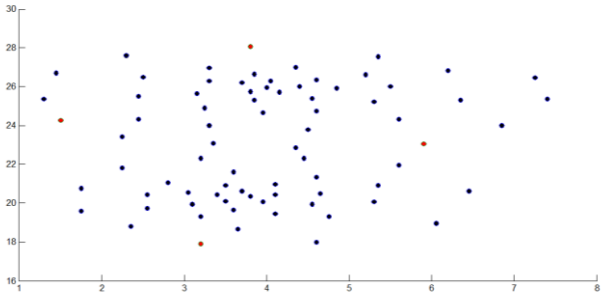

Fig 2. Object space with two touching clusters

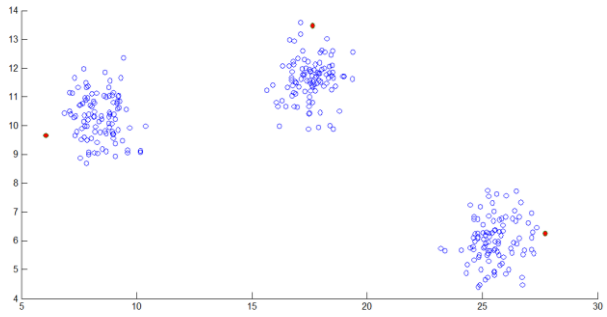

Fig 3. Object space with well separated clusters

Hence, instead of picking the first centre at random, we propose to pick the two initial centroids and then follow the D2 sampling for rest of the centroids. It can be combined with the MCMC technique to gain the required speed. The speed is decided by the length of Markov chain chosen. The two initial centroids should be picked as the two points farthest from each other so that it conforms to the basic idea of uniformly dividing the object space.

Remaining k-2 centres are picked according to the combined probability of distance from both centres. Applying laws of probability, the combined probability is

$$p(x) = p(x|c_1).p(x|c_2) = \frac{1}{2}\frac{d(x,c_1)^2}{\sum_{x' \in X} d(x',c_1)^2}.\frac{d(x,c_2)^2}{\sum_{x' \in X} d(x',c_2)^2} + \frac{1}{2}\frac{1}{|X|}$$

where, c_1 is the first centre and c_2 is the first centre We first multiply the distance factors and then add the normalizing term.

IV. Experiments and Results

The proposed initialization technique is tested through implementation in MATLAB along with other related initialization methods. The datasets used are the synthetic datasets popularly used to test output of clustering algorithms – A1, A2, A3 datasets, R15 dataset and D31 dataset. The chain length is kept as 10, 15 and 20. The quality of output is measured through value of objective function and accuracy of clustering.

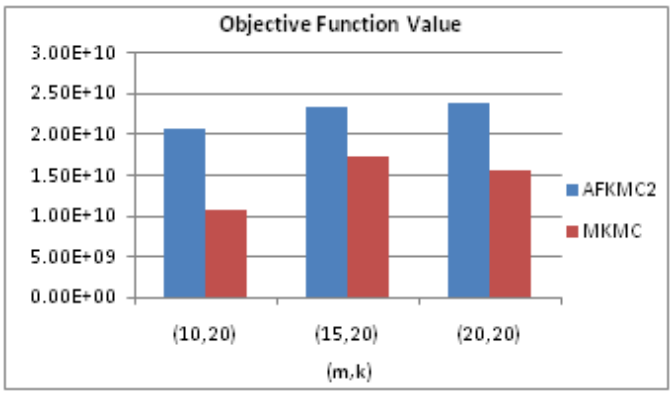

Fig 4. Objective function value of a1 data set

The objective function values are shown in Fig 4, 6, 8, 10 and 12; it can be noted that almost for every case the proposed has slightly larger value which can be tolerated as a trade-off for gain in classification accuracy. The number of misclassified objects is shown in Fig 5, 7, 9, 11 and 13. The number of Misclassifieds is very low for the proposed method as compared to afkmc2. Hence, the tradeoff is justified.

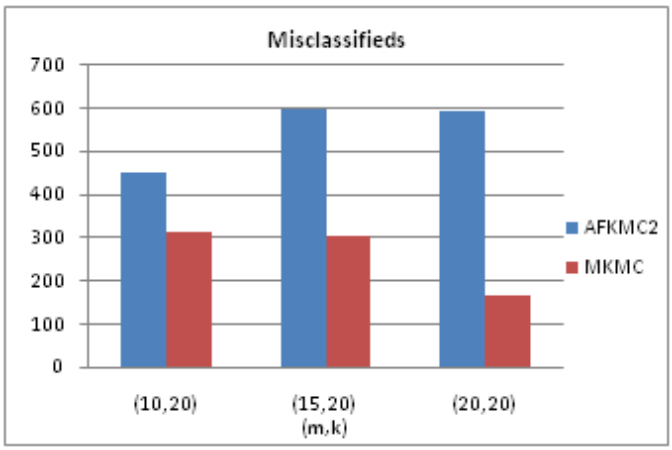

Fig 5. Misclassifieds of a1 data set

 ISSN: 2321-8169
89 - 93

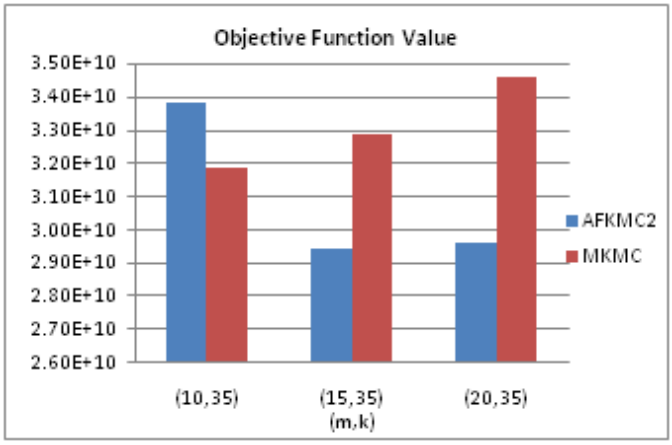

Fig 6. Objective function of a2 data set

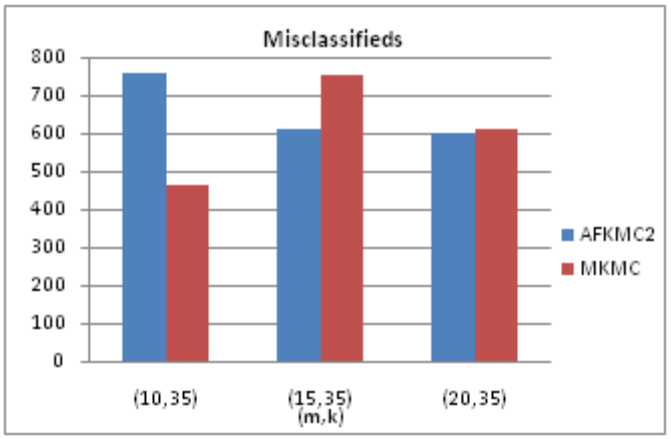

Fig 7. Misclassifieds of a2 data set

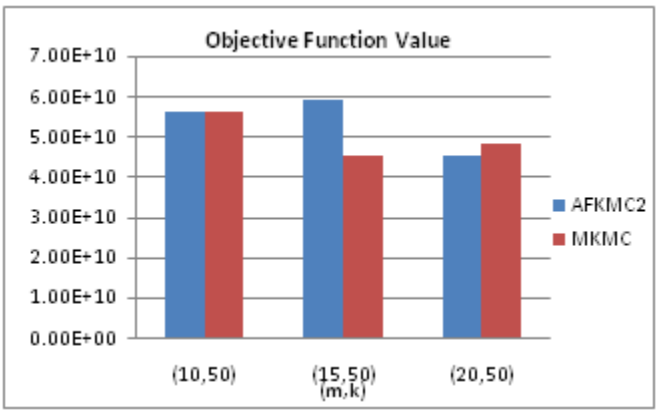

Fig 8. Objective function value of a3 data set

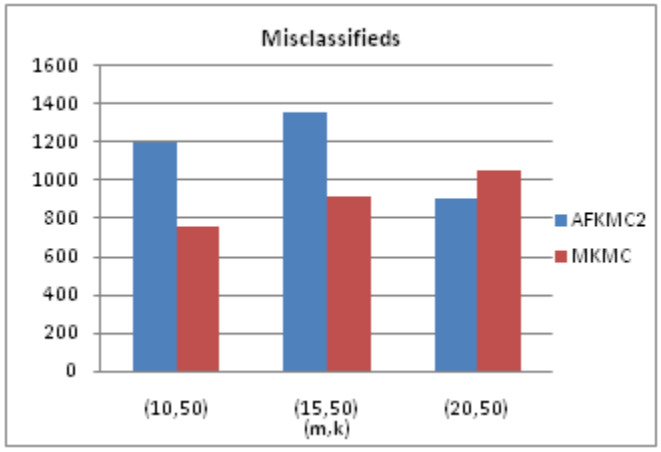

Fig 9. Misclassifieds of a3 data set

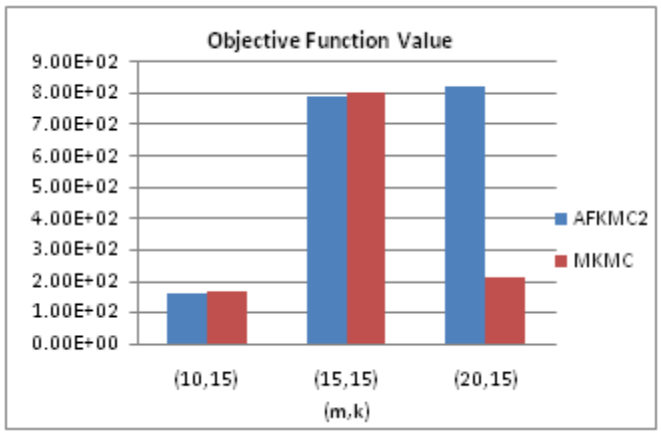

Fig 10. Objective function of R15 data set

V. CONCLUSION

Clustering through k-means is popular due to its simplicity and scalability. But it requires a good initialization to prevent from getting trapped into bad local minima. Kmeans++ is very popular and effective initialization based on distances among the points, but very time consuming process. Afkmc2 is a Monte Carlo sampling method that accelerates kmeans++ without sacrificing the quality of clusters. Yet, both have a drawback of needing a good point to start themselves. We propose to solve this issue by digressing a bit on time and improving quality in the trade-off. The method of sampling the points for initial seeds should begin with two points that are farthest from each other.

Such approach would improve kmeans++ too, though not analyzed in current work. For future we may consider our approach to test for improvement in all distance sampling based initialization methods of kmeans.

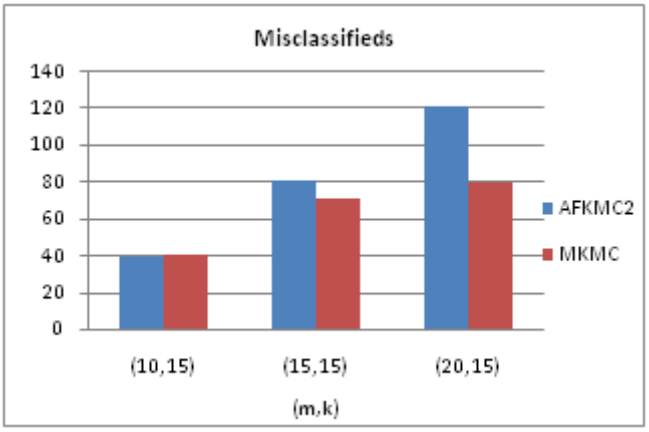

Fig 11. Misclassifieds of R15

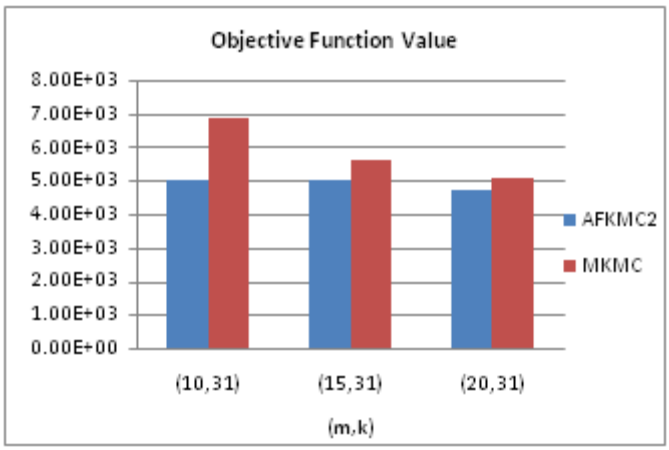

Fig 12. Objective function of D31

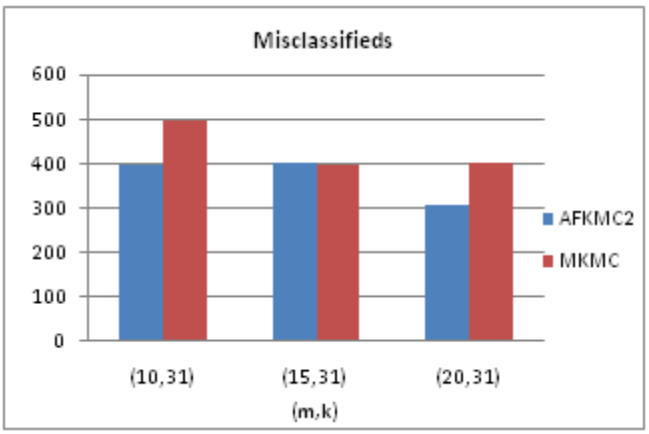

Fig 13. Misclassifieds of D31 data set

References

[1] S. Lloyd. Least squares quantization in pcm. IEEE Transactions on Information Theory, 28(2):129-137, March 1982.

[2] E. Forgy, Cluster analysis of multivariate data: efficiency vs. interpretability of classifications, Biometrics 21 (1965) 768.

[3] P. Bradley, U. Fayyad, Refining initial points for k-means clustering, in: Proceedings of the 15th International Conference on Machine Learning (ICML98), 1998, pp. 91–99.

[4] El Agha, Mohammed & Ashour, Wesam. (2012). Efficient and Fast Initialization Algorithm for K-means Clustering. International Journal of Intelligent Systems and Applications (IJISA). 4. 21-31. 10.5815/ijisa.2012.01.03.

[5] M. Al-Daoud and S. Roberts, "New methods for the initialization of clusters", Pattern Recogn. Lett. 17 (5), 451–455, 1994

[6] T. Su, J. Dy, A deterministic method for initializing K-means clustering, in: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, 2004, pp. 784–786.

[7] Katsavounidis, C.C. J. Kuo and Z. Zhang, "A New Initialization Technique for Generalized Lloyd Iteration", IEEE Signal Processing Letters 1 (10), 144–146, 1994

[8] T. Gonzalez, "Clustering to Minimize the Maximum Intercluster Distance", Theoretical Computer Science 38 (2–3), 293–306, 1985.

[9] D. Arthur and S. Vassilvitskii, "k-means++: The Advantages of Careful Seeding", Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete algorithms, pp. 1027-1035, 2007.

[10] Celebi, M Emre, Kingravi, Hassan A, and Vela, Patricio A. A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Systems with Applications, 40(1):200–210, 2013.

[11] O. Bachem, M. Lucic, S.H. Hassani and A. Krause, "Fast and Provably Good Seedings for k-Means", 30th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 2016.

[12] O. Bachem, M. Lucic, S.H. Hassani and A. Krause, "Approximate k-means++ in sublinear time", Conference on Artificial Intelligece (AAAI) , Feb 2016.