A Deterministic K-means Algorithm based on Nearest Neighbor Search

Omar Kettani, Benaissa Tadili, Faycal Ramdani
LPG Lab.
Scientific Institute
Mohamed V University, Rabat

ABSTRACT

In data mining, the k-means algorithm is among the most commonly and widely used method for solving clustering problems because of its simplicity and performance. However, one of the main drawback of this algorithm is that its accuracy and performance are sensitive to the initial choice of clustering centers, which are generated randomly. To overcome this drawback, we propose a simple deterministic method based on nearest neighbor search and k-means procedure in order to improve clustering results. Experimental results on various data sets reveal that the proposed method is more accurate than standard K-means algorithm.

General Terms

Clustering, Algorithms.

Keywords

Nearest Neighbor Search; Initial Centroid, K-means; Clustering Algorithm.

INTRODUCTION

Cluster analysis is widely used in various fields, including data analysis, biology, image processing, pattern recognition and machine learning. Clustering is the process of organizing data vectors into disjoint set called clusters such that the similarities among data members within the same cluster are maximal while similarities among data members from different clusters are minimal. The optimization of this criterion is a computationally NP hard problem in general Euclidean space *d*, even when the clustering process involves only two clusters [1].

Thus, many heuristic algorithms are generally used to find near optimal solution in reasonable computational time. One of the most widely used clustering methods is k-means clustering algorithm [2]. It is a relatively simple and efficient algorithm, but usually it converges to local optimum depending on initial cluster centroids, which are randomly generated. To overcome this drawback, authors proposed many initialization methods to improve the quality of clustering results. Bradley and Fayyad [3] proposed a refinement algorithm that builds a set of small random sub-samples of the data, then groups data in each sub-samples by K-means. Centroids of all sub-samples are then clustered together by K-means using the K centroids of each sub-sample as initial centers. Khan and Ahmad [4] described cluster center initialization algorithm (CCIA) based on considering values for each attribute of the given data set. This provided some information leading to a good initial cluster center. Arthur and Vassilvitskii [5] proposed k-means++, a careful seeding for initial cluster centers to improve clustering results. Recently, an initialization method for K-means algorithm using reverse nearest neighbor search and coupling degree was proposed by Ahmed and Ashour [6]. In [7], Zhang and Fang described an improved K-means clustering algorithm based on some core data point and a density threshold.

This paper suggests a deterministic approach (called KMNN) using nearest neighbor search for computing suitable initial clusters centroids instead of random ones, then apply k-means procedure to refine the clusters. Experiments are conducted on several data sets from UCI machine learning repository, in order to evaluate its performance.

In the following section we start with a brief description of the k-means algorithm and a formal definition of the clustering SSE error, then we describe the proposed KMNN algorithm. Section 3 describes a variant of the basic KMNN method which is slightly more accurate at the expense of requiring more computation. Section 4 reports our experimental results and comparisons with the original k-means algorithm. Finally section 5 provides conclusions and suggests directions for future research.

2.1. Brief review of K-Means clustering algorithm

Given a data set $X = \{x_1, ..., x_n\}$, $x_i \in R^d$ where each data point x_i corresponds to a vector of d attributes. The k-clustering problem aims at partitioning this data set into M disjoint subsets (or clusters) $C_1, ..., C_k$, such that clustering criterion is optimized. The most widely used clustering criterion is the sum of the squared Euclidean distances between each data point x_i and the centroid m_j (cluster center) of the subset C_j which contains x_i. This criterion is called clustering error and depends on the cluster centers $c_1, ..., c_k$:

$$SSE = \sum_{j=1}^{k} \sum_{x_i \in C_j} \| x_i - m_j \|^2$$

Where $\|.\|^2$ denotes the Euclidean norm in R^d,

$$c_j = \sum_{x_i \in C_j} x_i / |C_j|$$

denotes the centroids of cluster C_j and $|C_j|$ denotes the number of instances in C_j.

The K-means algorithm has the following steps:

Algorithm KM

Input: data set $X=\{x_1,..., x_n\}$, $x_i \in R^d$

a positive integer $k<n$

Output: k mutually disjoint clusters $C_1,..., C_k$ such that $C_1 \cup,...,\cup C_k=X$

Step 1: Select k initial cluster centers $c_1, c_2,...,c_k$ randomly from the given n points X.

Step 2: Assign each point x_i, i =1, 2, ... , n to the cluster C_j corresponding to the cluster center c_j, for j =1, 2, ... , k iff

$\|x_i - c_j\| \le \|x_i - c_p\|$, p = 1, 2, ..., k and $j \neq p$

Step 3: Compute new cluster centers $nc_1, nc_2,...,nc_k$ as follows:

$nc_j = \sum x_i / |C_j|$ for j = 1, 2, ..., k.

$x_i \in C_j$

Step 4: If $nc_i = c_i$, $\forall$ i = 1, 2, ..., k, then terminate. Otherwise continue from step 2.

2.2 The proposed method

Instead of choosing the k initial cluster centers $c_1, c_2,...,c_k$ randomly from the given data set X, the proposed KMNN method picks the first point in X, then computes its $\lceil n/k \rceil -1$ nearest neighbors which constitute the first cluster C_1 whose centroid is set to c_1, then C_1 is deleted from X. This process is repeated k times until the k initial cluster centers $c_1, c_2,...,c_k$ are assigned. After that, the K-means algorithm is applied to refine the clusters. The proposed KMNN algorithm is outlined below:

Algorithm KMNN

Input: data set $X=\{x_1,..., x_n\}$, $x_i \in R^d$

a positive integer $k<n$

Output: k mutually disjoint clusters $C_1,..., C_k$ such that $C_1 \cup,...,\cup C_k=X$

Step 1:

For j=1 to k do

$C_j \leftarrow$ NNsearch(x_1, X , $\lceil n/k \rceil -1$) $\cup \{x_1\}$

$c_j \leftarrow \sum x_i / \lceil n/k \rceil$

$x_i \in C_j$

$X \leftarrow X - C_j$

EndFor

Step 2: Assign each point x_i, i =1, 2, ... , n to the cluster C_j corresponding to the cluster center c_j, for j =1, 2, ... , k iff

$\|x_i - c_j\| \le \|x_i - c_p\|$, p = 1, 2, ..., k and $j \neq p$

Step 3: Compute new cluster centers $nc_1, nc_2,...,nc_k$ as follows:

$nc_i = \sum x_i / |C_j|$ for j= 1, 2, ..., k.

$x_i \in C_j$

Step 4: If $nc_i = c_i$, $\forall$ i = 1, 2, ..., k, then terminate. Otherwise continue from step 2.

Clearly, the performance of this method depends on the complexity of the NNsearch procedure and the complexity of K-Means algorithm used. Therefore, in order to speedup this method, one can use a fast implementation of Nearest Neighbor Search algorithm like a method described in [9] and implement a fast version of K-Means algorithm like those described in [10] and [11].

3. A variant of the proposed method

If the given data set X has moderate size, one can use the following variant of the basic KMNN method (called KMNN') which is slightly more accurate than former method, but requires more computation effort aiming to minimize the SSE clustering criterion by using an additional inner loop.

The pseudo code of KMNN' algorithm is outlined below:

Algorithm KMNN'

Input: data set $X=\{x_1,..., x_n\}$, $x_i \in R^d$

a positive integer $k<n$

Output: k mutually disjoint clusters $C_1,..., C_k$ such that $C_1 \cup,...,\cup C_k=X$

Step 1:

For j=1 to k do

For h=1 to | X | do

$C_h \leftarrow$ NNsearch(x_h, X , $\lceil n/k \rceil -1$) $\cup \{x_h\}$

$c_h \leftarrow \sum x_i / \lceil n/k \rceil$

$x_i \in C_h$

$d_h \leftarrow \sum \|x_i - c_h\|^2$

$x_i \in C_h$

EndForh

$m \leftarrow \underset{1 \le i \le |X|}{\text{ArgMin}}(d_i)$

$C_j \leftarrow C_m$

$c_j \leftarrow c_m$

$X \leftarrow X - C_j$

EndForj

Step 2: Assign each point x_i, i =1, 2, ... , n to the cluster C_j corresponding to the cluster center c_j, for j =1, 2, ... , k iff

$\| x_i - c_j \| \leq \| x_i - c_p \|$, p = 1, 2, ..., k and j≠p

Step 3: Compute new cluster centers $nc_1, nc_2,...,nc_k$ as follows:

$$nc_i = \sum x_i / |C_j| \quad \text{for j= 1, 2, ..., k.}$$
$$x_i \in C_j$$

Step 4: If $nc_i = c_i$, ∀ i = 1, 2, ..., k, then terminate. Otherwise continue from step 2.

4 Experimental evaluation

In order to evaluate the proposed clustering algorithm, experiment was conducted on several data sets from UCI machine learning repository [12]. The initial centroid for standard k-means algorithm is selected randomly. The experiment was conducted 10 times for different sets of values of the initial centroids, which were selected randomly. In each experiment, the SSE and silhouette value was computed and taken the average of all experiments.

The silhouette function [8] provides a measure of the quality of the separation between the clusters obtained by using the K-means algorithm. In an object i belonging to the cluster C_k , the average dissimilarity of i to all other objects of C_k is denoted by a_k (i). Analogously, in cluster C_j, the average dissimilarity of i to all objects of C_j is called dis(i,C_j). After computing dis(i,C_j) for all clusters $C_j \neq C_k$, the smallest one is selected as follows,

a_j (i) = min{dis(i,C_j)}, ∀ j such that $C_j \neq C_k$.

This value represents the dissimilarity of the object i to its neighbor cluster. Thus, the silhouette values, silh(i) are given by the following equation:

silh(i) =(a_k (i) − a_j (i))/max{ a_k (i), a_j (i)}

The silh(i) can vary between −1 and +1, +1 denotes clear cluster separation and −1 marks points with *bad* cluster assignment. The objective function is the average of silh(i) over the number of objects to be classified, and the best clustering is reached when the above mentioned function is maximized.

In our experiments we used MATLAB software [13] and Windows 7 with Intel Core 2Duo CPU 2.8 GHZ with RAM 4.0GB.

The SSE results and average of silhouette values on 12 UCI data sets are reported in table 1 and some clustering results are shown in Fig. 1 to 6.

From these measurements we can see that the proposed KMNN algorithm outperform the random initialization KM algorithm in most cases.

We have also evaluated the proposed variant algorithm KMNN' on the same data sets, and we noticed that this variant performs slightly better than KMNN only on 2 data set among the twelve tested data sets as shown in table 2.

Table 1: SSE results and average silhouette values on various data sets using KM and KMNN respectively.

Data set	Algorithm KM		Algorithm KMNN	
	SSE	SIL	SSE	SIL
Glass	1.0967e+004	0.6771	1.0928e+004	0.6802
S1	1.9534e+013	0.7701	1.4744e+013	0.8121
S2	1.3279e+013	0.8008	1.3279e+013	0.8009
S3	1.9334e+013	0.6364	1.8787e+013	0.6412
S4	1.6740e+013	0.6268	1.5704e+013	0.6447
Ruspini	4.8309e+004	0.7024	1.29E+004	0.9086
DIM032	1.2973e+007	0.7342	2.3254e+005	0.9962
R15	2.0950e+003	0.6954	109.8706	0.9361
A1	2.0053e+010	0.6802	1.2146e+010	0.7892
Aggregation	1.4651e+004	0.6317	1.1111e+004	0.6717
Compound	4.9229e+003	0.5033	4.7323e+003	0.5686
Yest	52.51	0.2431	46.1477	0.2662

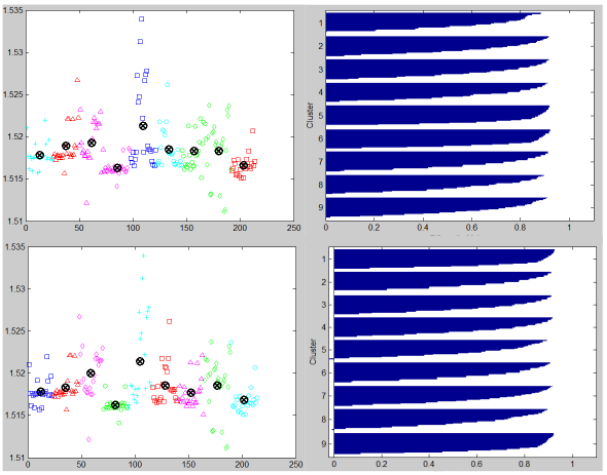

Fig. 1: clustering results of glass data set using KM and KMNN respectively and their corresponding silhouette plots.

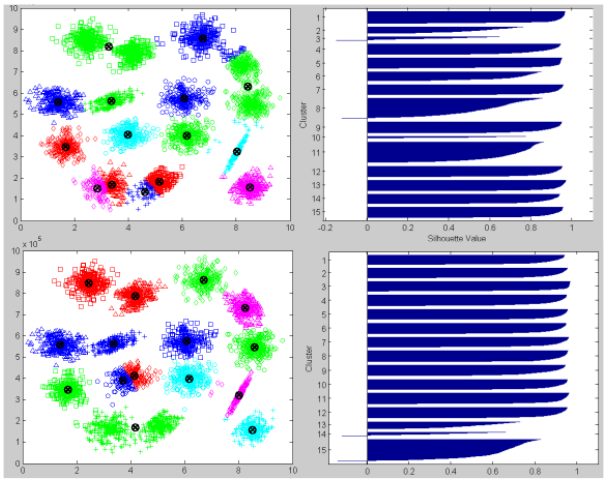

Fig. 2: clustering results of S1 data set using KM and KMNN respectively.

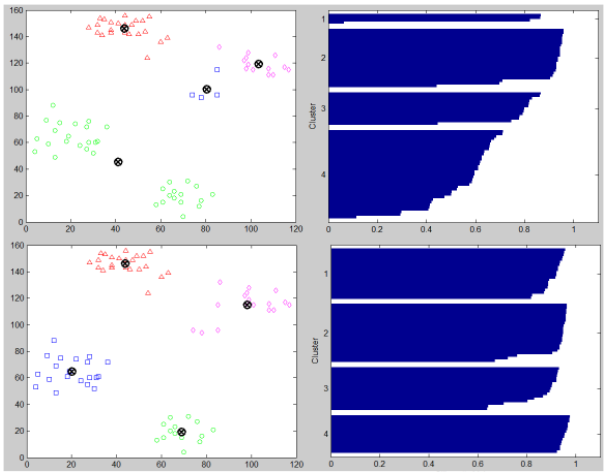

Fig. 3: clustering results of Ruspini data set using KM and KMNN respectively.

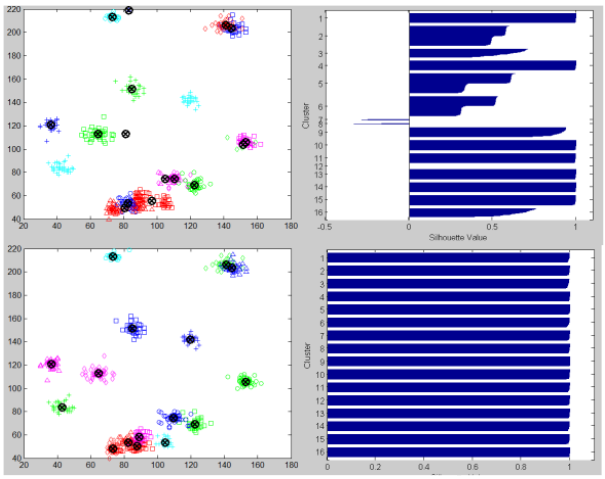

Fig. 4: clustering results of DIM032 data set using KM and KMNN respectively.

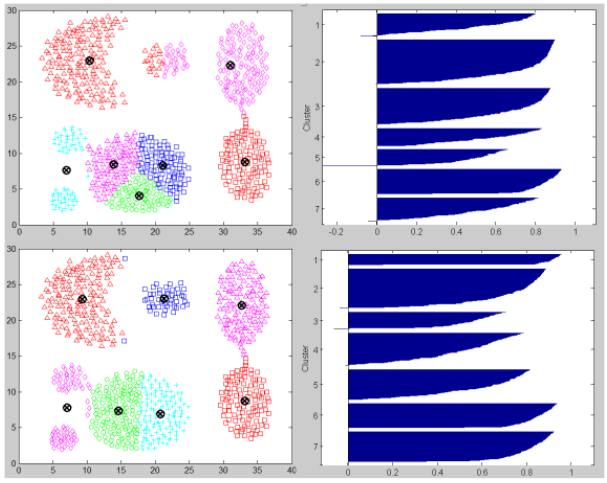

Fig. 5: clustering results of Aggregation data set using KM and KMNN respectively.

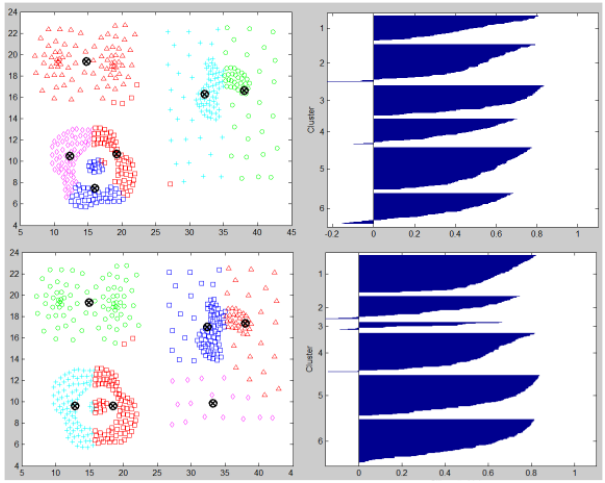

Fig. 6: clustering results of Compound data set using KM and KMNN respectively.

Table 2: SSE results and average silhouette values on two data sets where KMNN' performs better than KMNN.

Data set	Algorithm KM		Algorithm KMNN'	
	SSE	SIL	SSE	SIL
Aggregation	1.1111e+004	0.6717	1.1109e+004	0.6718
Compound	4.7323e+003	0.5686	3.9290e+003	0.6018

5. CONCLUSION

In this paper a new approach for the initialization of the K-means algorithm has been proposed which is based on the nearest neighbor search procedure as a preprocessing step. Experiments conducted on both synthetic and real data sets, showed improvement in accuracy of the clustering results. We have also evaluated a variant algorithm which spent more computation time in order to minimize earlier the SSE

criterion, but we noticed surprisingly that this variant performs slightly better than KMNN only on two data sets among twelve tested data sets. Thus the proposed approach is a good compromise between efficiency and accuracy. Also, it is simple and easy to implement.

As future work, we intend to apply Principal Component Analysis on original data as a possible way to improve both performance and accuracy of the proposed method. We also intend to investigate a possible modification of this method aiming at finding a parameter-free algorithm which will automatically detect the optimal number of clusters of a given data set.

6. REFERENCES

[1] Aloise, D.; Deshpande, A.; Hansen, P.; Popat, P. (2009). "NP-hardness of Euclidean sum-of-squares clustering". Machine Learning 75: 245–249. doi:10.1007/s10994-009-5103-0.

[2] Lloyd., S. P. (1982). "Least squares quantization in PCM". IEEE Transactions on Information Theory 28 (2): 129–137. doi:10.1109/TIT.1982.1056489.

[3] P.S. Bradley and U.M. Fayyad, "Refining initial points for K-means Clustering", Proceeding of The Fifteenth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1998, pp. 91-99.

[4] Khan and A. Ahmad, "Cluster Center Initialization for K-mean Clustering", Pattern Recognition Letters, Volume 25, Issue 11, 2004, pp. 1293-1302

[5] Arthur, D. and S. Vassilvitskii, 2007. K-means++: The advantages of careful seeding. Proceeding of the 18th Annual ACM-SIAM Symposium of Discrete Analysis, Jan. 7-9, ACM Press, New Orleans, Louisiana, pp:1027-1035.

[6] Ahmed and W. Ashour "An Initialization Method for the K-means Algorithm using RNN and Coupling Degree" International Journal of Computer Applications (0975 – 8887) Volume 25– No.1, July 2011

[7] C. Zhang and Z. Fang "An Improved K-means Clustering Algorithm" Journal of Information & Computational Science 10: 1 (2013) 193–199

[8] L. Kaufman and P. J. Rousseeuw. Finding groups in Data: "an Introduction to Cluster Analysis". Wiley, 1990.

[9] Yoonho Hwang; Bohyung Han; Hee-Kap Ahn "A fast nearest neighbor search algorithm by nonlinear embedding" Computer Vision and Pattern Recognition (CVPR), 2012 IEEE

[10] Ming-Chao Chiang, Chun-Wei Tsai, Chu-Sing Yang "A time-efficient pattern reduction algorithm for k-means clustering" Information Sciences 181 (2011) 716–731

[11] You Li, Kaiyong Zhao, Xiaowen Chu , Jiming Liu "Speeding up k-Means algorithm by GPUs" Journal of Computer and System Sciences 79 (2013) 216–229

[12] Merz C and Murphy P, UCI Repository of Machine Learning ftp://ftp.ics.uci.edu/pub/machine-Learning-databases Clustering datasets: http://cs.joensuu.fi/sipu/datasets/

[13] http://www.mathworks.com