RESEARCH ARTICLE OPEN ACCESS

Optimising Data Using K-Means Clustering Algorithm

Mubeena Shaik[1], Naseema Shaik[2], Ahmed Unnisa Begum[1], Sameera Iqbal[2], Dr P.Uma[3], Raheem Unnisa Begum[2]
[1]Lecturer, Jazan University, Saudi Arabia.
[2]Lecturer, Kind Khalid University, Saudi Arabia.
[3]Assist. Professor, Jazan University, Saudi Arabia.

Abstract
K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other.

I. Introduction

The Clustering is the process of grouping physical or abstract objects into classes of similar objects. And also is a process of partitioning a set of data (or objects) into a set of meaningful sub-classes, called clusters.

Whereas K-means clustering is a method often used to partition a data set into k groups. It proceeds by selecting k initial cluster and then iteratively refining them as follows:

1. Initialize the center of the clusters
   μi= some number ,i=1,...,k
2. Attribute the closest cluster to each data point
   ci={ j:d (xj,μi) ≤ d (xj,μl) , l≠i , j=1 ,..., n }
3. Set the position of each cluster to the mean of all data points belonging to that cluster
   $\mu i= 1 |ci| \sum j \in ci\ xj , \forall i$
4. Repeat steps 2-3 until convergence therefore |c|= number of elements in c.

This algorithm eventually converges to a point, although it is not necessarily the minimum of the sum of squares. That is because the problem is non-convex and the algorithm is just a heuristic, converging to a local minimum. The algorithm stops when the assignments do not change from one iteration to the next iteration.

If the choice of the data is incorrect, the process becomes invalidate. Where as the best number of clusters is to try K-means clustering with different number of clusters and measure the resulting sum of squares[6].

Cluster analysis [4] is one of the major data analysis methods which is widely used for many practical applications in emerging areas like Bioinformatics [5]. Clustering is the process of partitioning a given set of objects into disjoint clusters. This is done in such a way that objects in the same cluster are similar while objects belonging to different clusters differ considerably, with respect to their attributes.

Regarding computational complexity, finding the optimal solution to the *k*-means clustering problem for observations in *d* dimensions is:

- NP-hard in general Euclidean space *d* even for 2 clusters [7]
- NP-hard for a general number of clusters *k* even in the plane [8]
- If *k* and *d* (the dimension) are fixed, the problem can be exactly solved in time $O(n^{dk+1} \log n)$, where *n* is the number of entities to be clustered [9]

II. Clustering by K-means Algorithm :

This section describes the original k-means clustering algorithm. The idea is to classify a given set of data into k number of disjoint clusters, where the value of k is fixed in advance. The algorithm consists of two separate phases: the first phase is to define k centroids, one for each cluster. The next phase is to take each point belonging to the given data set and associate it to the nearest centroid. Euclidean distance is generally considered to determine the distance between data points and the centroids. When all the points are included in some clusters, the first step is completed and an early grouping is done. At this point we need to recalculate the new centroids, as the inclusion of new points may lead to a change in the cluster centroids. Once we find k new centroids, a new binding is to be created between the same data points and the nearest new centroid, generating a loop. As a result of this loop, the k centroids may change their position in a step by step manner. Eventually, a situation will be reached where the centroids do not move anymore. This signifies the convergence criterion for clustering.

Pseudocode for the k-means clustering algorithm is listed as Algorithm 1 [1].

→ The K- means clustering Algorithm.

Input:
D = {d1, d2,.....,dn} //set of n data items. k // Number of desired clusters
Output:
A set of k clusters.
Steps:
1. Arbitrarily choose k data-items from D as initial centroids;
2. Repeat
3. Assign each item d_i to the cluster which has the closest centroid;
Calculate new mean for each cluster;
Until convergence criteria is met.

The k-means algorithm is the most extensively studied clustering algorithm and is generally effective in producing good results. The major drawback of this algorithm is that it produces different clusters for different sets of values of the initial centroids. Quality of the final clusters heavily depends on the selection of the initial centroids. The k-means algorithm is computationally expensive and requires time proportional to the product of the number of data items, number of clusters and the number of iterations.

In the optimization of clustering method discussed in this paper, both the phases of the original k-means algorithm are modified to improve the accuracy and efficiency. The enhanced method is described in Algorithm 2.

Algorithm – 2 : The optimized Method

Input:
D = {d1, d2,.....,dn} // set of n data items k // Number of desired clusters
Output:
A set of k clusters.
Steps:
1: Determine the initial centroids of the clusters by using Algorithm 2.1.
2: Assign each data point to the appropriate clusters by using Algorithm 2.2.

In the first point, the initial centroids are determined systematically so as to produce clusters with better accuracy [3]. The second point makes use of a variant of the clustering method discussed in [2]. It starts by forming the initial clusters based on the relative distance of each data-point from the initial centroids. These clusters are subsequently fine-tuned by using a heuristic approach, thereby improving the efficiency. The two points of the enhanced method are described below as Algorithm 2.1 and Algorithm 2.2.

Algorithm - 2.1 Finding the initial centroids

Input:
D = {d1, d2,.....,dn} // set of n data items k // Number of desired clusters
Output:
A set of k initial centroids .
Steps: 1. Set m = 1;
2. Compute the distance between each data point and all other data- points in the set D;
3. Find the closest pair of data points from the set D and form a data-point set Am (1<= m <= k) which contains these two data- points, Delete these two data points from the set D;
4. Find the data point in D that is closest to the datapoint set Am, Add it to Am and delete it from D;
5. Repeat step 4 until the number of data points in Am reaches 0.75*(n/k);
6. If m<=m<=k) find the arithmetic mean of the vectors of data points in Am, these means will be the initial centroids.

Algorithm 2.1 describes the method for finding initial centroids of the clusters [3]. Initially, compute the distances between each data point and all other data points in the set of data points. Then find out the closest pair of data points and form a set A1 consisting of these two data points, and delete them from the data point set D. Then determine the data point which is closest to the set A1, add it to A1 and delete it from D. Repeat this procedure until the number of elements in the set A1 reaches a threshold. At that point go back to the second step and form another data-point set A2. Repeat this till 'k' such sets of data points are obtained. Finally the initial centroids are obtained by averaging all the vectors in each data-point set. The Euclidean distance is used for determining the closeness of each data point to the cluster centroids. The distance between one vector X = (x1, x2, ...xn) and another vector Y = (y1, y2, …….yn) is obtained as

$$d(X,Y)=\sqrt{(x1-y1)2}+(x2-y2)2+\cdots\ldots(xn-yn)2$$

The distance between a data point X and a data-point set D is defined as d(X, D) = min (d (X, Y), where Y ∈ D). The initial centroids of the clusters are given as input to the second phase, for assigning data-points to appropriate clusters. The steps involved in this phase are outlined as Algorithm 2.2

Algorithm 2.2 Assigning data-points to clusters

Input:
D = {d1, d2,.....,dn} // set of n data-points.
C = {c1, c2,.....,ck} // set of k centroids

Output:
A set of k clusters
Steps:
1. Compute the distance of each data-point di (1<=i<=n) to all the centroids cj (1<=j<=k) as d(di, cj);
2. For each data-point di, find the closest centroid cj and assign di to cluster j.
3. Set ClusterId[i]=j; // j:Id of the closest cluster
4. Set Nearest_Dist[i]= d(di, cj);
5. For each cluster j (1<=j<=k), recalculate the centroids;
6. Repeat
7. For each data-point d_i,
7.1 Compute its distance from the centroid of the present nearest cluster;
7.2 If this distance is less than or equal to the present nearest distance, the data-point stays in the cluster; Else
7.2.1 For every centroid cj (1<=j<=k) Compute the distance d(di, cj); Endfor;
7.2.2 Assign the data-point di to the cluster with the nearest centroid cj
7.2.3 Set ClusterId[i]=j;
7.2.4 Set Nearest_Dist[i]= d(di, cj); Endfor;
8. For each cluster j (1<=j<=k), recalculate the centroids; Until the convergence criteria is met.

The first step in Phase 2 is to determine the distance between each data-point and the initial centroids of all the clusters. The data-points are then assigned to the clusters having the closest centroids. This results in an initial grouping of the data-points. For each data-point, the cluster to which it is assigned (ClusterId) and its distance from the centroid of the nearest cluster (Nearest_Dist) are noted. Inclusion of data-points in various clusters may lead to a change in the values of the cluster centroids. For each cluster, the centroids are recalculated by taking the mean of the values of its data-points. Up to this step, the procedure is almost similar to the original k-means algorithm except that the initial centroids are computed systematically.

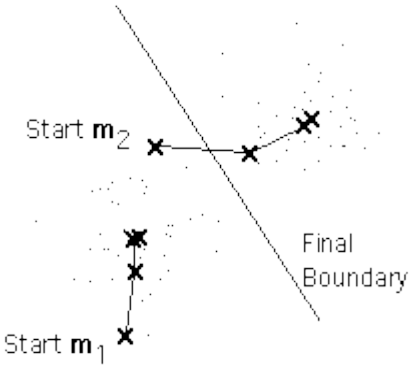
III. An Example

Suppose that we have n sample feature vectors x_1, x_2, ..., x_n all from the same class, and we know that they fall into k compact clusters, k < n. Let m_i be the mean of the vectors in cluster i. If the clusters are well separated, we can use a minimum-distance classifier to separate them. That is, we can say that x is in cluster i if || x - m_i || is the minimum of all the k distances. This suggests the following procedure for finding the k means:

- Make initial guesses for the means m_1, m_2, ..., m_k
- Until there are no changes in any mean
  - Use the estimated means to classify the samples into clusters
  - For i from 1 to k
- Replace m_i with the mean of all of the samples for cluster i
  - end_for

- end_until

Here is an example showing how the means m_1 and m_2 move into the centers of two clusters.



A simple approach is to compare the results of multiple runs with different k classes and choose the best one according to a given criterion ,Note that we need to be careful because increasing k results in smaller error function values by definition, but also an increasing risk of overfitting.

IV. Conclusion

The k-means algorithm is widely used for clustering to optimize large sets of data. But the standard algorithm do not always guarantee good results as the accuracy of the final clusters depend on the selection of initial centroids. Moreover, the computational complexity of the standard algorithm is objectionably high owing to the need to reassign the data points a number of times, during every iteration of the loop. This paper presents an opimized k-means algorithm which combines a systematic method for finding initial centroids and an efficient way for assigning data points to clusters. This method ensures the entire process of clustering in O(n2) time without sacrificing the accuracy of clusters. The previous improvements of the k-means algorithm compromise on either accuracy or efficiency.

References:

[1] Margaret H. Dunham, Data Mining-Introductory and Advanced Concepts, Pearson Education, 2006.

[2] Chaturvedi J. C. A, Green P, "*K-modes clustering*," J. Classification, (18):35–55, 2001.

[3] Yuan F, Meng Z. H, Zhang H. X and Dong C. R, "*A New Algorithm to Get the Initial Centroids*," Proc. of the 3rd International Conference on Machine Learning and Cybernetics, pages 26–29, August 2004.

[4] Jiawei Han M. K, Data Mining Concepts and Techniques, Morgan Kaufmann Publishers, An Imprint of Elsevier, 2006.

[5] Amir Ben-Dor, Ron Shamir and Zohar Yakini, "*Clustering Gene Expression Patterns*," Journal of Computational Biology, 6(3/4): 281-297, 1999

[6] Daxin Jiang, Chum Tong and Aidong Zhang, "*Cluster Analysis for Gene Expression Data*," IEEE Transactions on Data and Knowledge Engineering, 16(11): 1370-1386, 2004.

[7] 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

[8] Mahajan, M.; Nimbhorkar, P.; Varadarajan, K. (2009). "*The Planar k-Means Problem is NP-Hard*". Lecture Notes in Computer Science 5431: 274–285. doi:10.1007/978-3-642-00202-1_24

[9] Inaba, M.; Katoh, N.; Imai, H. (1994). Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. Proceedings of 10th ACM Symposium on Computational Geo metry. pp. 332–339. doi:10.1145/177424.178042