ACADEMIC World Academic Union

ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 9, No. 4, 2014, *pp*. 277-284

Enhanced *K*-Means Clustering Algorithm using A Heuristic Approach

Vighnesh Birodkar [1] and Damodar Reddy Edla [1,*]

[1] Department of Computer Science and Engineering
National Institute of Technology Goa – 403 401, India.
Email: vighneshbirodkar@gamil.com, dr.reddy@nitgoa.ac.in
(*Received March 31, 2014, accepted October 1, 2014*)

Abstract. *K*-means algorithm is one of the most popular clustering algorithms that has been survived for more than 4 decades. Despite its inherent flaw of not knowing the number of clusters in advance, very few methods have been proposed in the literature to overcome it. The paper contains a fast heuristic algorithm for guessing the number of clusters as well as cluster center initialization without actually performing *K*-means, under the assumption that the clusters are well separated in a certain way. The proposed algorithm is experimented on various synthetic data. The experimental results show the effectiveness of the proposed approach over the existing.

Keywords: partitional clustering, K-means, unsupervised learning, cluster center, synthetic data

1. Introduction

Clustering is a widely used approach in data-mining. It has a variety of application, including Medicine, [1],Feature Detection [2] ,Geology [3] and Robotics [4].The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering[5] . Clustering is broadly divided into partitioning, hierarchical, density-based, grid-based, model-based and constraint-based methods.

Partitioning method divides *n* objects into *k* groups such that each group has at least one object and each object belongs to only one group [5]. *K*-Means is one of the most widely used partition-based clustering algorithms due to its simplicity. It divides a set of objects into a fixed number of groups called clusters. Let $P = \{p_i \mid 1 \le i \le N\}$ be a set of objects. *K*-Means divides *P* into *K* clusters. If $C = \{C_i \mid 1 \le i \le K\}$ is a set of *K* centers *K*-Means tries to minimize the sum of Euclidean distances of objects from its cluster centers, which can be given by

$$sum = \sum_{i=1}^{n} dist\left(x_i - c_k\right)$$

where $dist(p_i, c_k)$ is the Euclidean distance of object p_i from its cluster center. *K*-Means attempts to find a local minima for sum, and hence requires that *K* (number of clusters) to be known in advance, the global minima being $sum = 0$ when $K = N$. The problem of finding clusters so as to minimize the Euclidean Sum of Squares is *NP*-Hard [6] and *K*-Means suffices only as a heuristic

2. A Review

K-means is the most popular clustering technique of this model developed by MacQueen [6] in 1967. However, it is sensitive to the random selection of initial cluster centers. In addition to that, a prior knowledge of the number of clusters is necessity to input to K-means. Many researches proposed various methods [7], [8] to overcome these problems. Kanungo et al. [9] proposed a novel initialization method for K-means using kd-tree. This scheme does not pass information from one stage to its next. Du et al. [10] developed an initialization scheme for K-means clustering called *PK*-means to cluster the gene expression data. The convergence rate of this technique is fast and the computational load is less. A novel clustering algorithm called modified filtering algorithm (MFA) has been proposed in [11]. It is the improvement of the algorithm in [12]. A fast *K*-means clustering algorithm named FKMCUCD was proposed in [13] using cluster center displacement. This method is significant for high-dimensional large data. Zalik [14] proposed an efficient algorithm named *K*´-means to enhance the K-means algorithm by exploiting a cost function. This scheme fails when the clusters are of various shapes such as elliptical. Redmond et al. [15] proposed a novel seed selection algorithm using kd-tree [16]. This scheme is unable to deal with

the noise. Cao et al. [17] proposed an algorithm by defining the cohesion degree of the neighborhood of a given point and the coupling degree between neighborhoods of the points. This algorithm has quadratic time complexity. Khan et al. [18] designed an algorithm called CCIA. This method first develops $k'(>k)$ cluster centers from which the desired k centers are chosen. Lu et al. [19] contributed with a hierarchical initialization approach in which the clustering problem has treated as a weighted clustering problem. A genetic clustering algorithm named GAGR [20] has been proposed to cluster the genome data using *K*-means. It uses the genetic algorithm with gene rearrangement process. Ahmad et al. [21] proposed an enhanced K-means clustering algorithm for mixed numeric and categorical data based on co-occurrence of the values. An algorithm called KGA [22] was proposed using the genetic algorithm. This method may not produce fine results whenever the number of clusters is unknown. An improved version of K-means called *K**-means has been developed in [23]. It is unable to deal with the noisy data. Likas et al. [24] proposed a global *K*-means clustering algorithm in which the clusters are formed using a global search procedure. A recursive method is proposed by Duda and Hart [25]. Milligan [26] developed an enhanced algorithm based on Ward's hierarchical method [27] that helps in finding the initial cluster centers. The algorithm proposed by Fisher [28] generates good seeds by constructing initial hierarchical clustering based on [29]. Both Higgs et al., [30] and Snarey et al. [31] developed a method using MaxMin algorithm to choose a subset of the original database as initial cluster centers. Bradley et al., [32] formed the initial clusters based on the bilinear program. Tou and Gonzales [33] presented a method which entirely depends on the order of the points and the threshold value. Linde et al., [34] proposed a method based on Binary Splitting (BS). Here, the clusters quality depends on the selection of a random vector. Kaufman and Rousseeuw [35] developed a method based on the reduction in the Distortion. Babu and Murty [36] proposed a technique for the near optimal seed selection based on genetic programming. This is not robust for large data bases. Huang and Harris [37] projected a method called Direct Search Binary Splitting (DSBS) based on the Principal Component Analysis (PCA) and the vector of Linde et al., [35]. Thiesson et al., [38] designed an algorithm that depends on the mean value of the given data. Bradley and Fayyad [39] proposed an initialization approach for K-means using the Forgy's method [40].

3. Proposed Algorithm

Let P = $\{p_i \mid 1 \le i \le N\}$ be a set of *N* objects. Let *ClusterSet* = $\{C_i \mid 1 \le i \le K\}$ be the set of K desired Clusters.

$Let\ x \in C_i \text{ and } y \in C_j. \quad dist(x,y) \mid i = j,\ \forall x\ \forall y < \text{dist}(x,y) \mid i \neq j,\ \forall x\ \forall y$

The algorithm first finds *K* objects from *K* different clusters. Assume that at a certain stage *ClusterPoints* is a set of *m* objects (where $m < K$) from *m* different clusters. To find the $(m + 1)^{th}$ object we find the Minimum Euclidean Distance of all objects in P from any object in *ClusterPoints* and assign it as *minDist*. Under the assumption that the clusters are Well-separated all the objects belonging to the same clusters as any object in *ClusterPoints* will have lesser *minDist* than an object not in the same cluster as any object in *ClusterPoints*. If we select the object q with the Maximum *minDist* it is assured that it does not belong to the same cluster as any of the objects in *ClusterPoints*. We add the *q* in *ClusterPoints* and increment *m* by 1 and the process is repeated. To start the algorithm a random object is chosen and added to *ClusterPoints* and m is set to 1.The algorithm thus assures that the first *K* objects chosen are from different clusters.

Upon successive iterations of the above process there will be a state where $m > K$, particularly $m = K + 1$. We need to identify this state for the algorithm to stop and return *K*. To achieve this the proposed algorithm relies on a heuristic based on Euclidean Distance. If 2 objects belong to the same they will have similar Euclidean Distances from objects from other clusters. Algorithm 1 takes 2 arrays of length *N* and outputs a real number which is significantly lower for arrays of Euclidean Distances of two points belonging to the same cluster. At each iteration the algorithm calls Algorithm 1 and halts when the value is less than R times all the previously computed values by Algorithm 1

Diff Algorithm:

Input
A – Array of objects
B - Array of objects
N – Length of Arrays

Output
ans – A Positive Real Number
sum = 0
for $i = 1 \rightarrow N$

$$sum = sum + \left(\frac{2*(A[i]-B[\text{i}])}{A[i]+B[i]} \right)^2$$

end