Expert Systems with Applications 40 (2013) 200–210

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A comparative study of efficient initialization methods for the k-means clustering algorithm

M. Emre Celebi [a,*], Hassan A. Kingravi [b], Patricio A. Vela [b]

[a] Department of Computer Science, Louisiana State University, Shreveport, LA, USA
[b] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

ARTICLE INFO

Keywords:
Partitional clustering
Sum of squared error criterion
k-means
Cluster center initialization

ABSTRACT

K-means is undoubtedly the most widely used partitional clustering algorithm. Unfortunately, due to its gradient descent nature, this algorithm is highly sensitive to the initial placement of the cluster centers. Numerous initialization methods have been proposed to address this problem. In this paper, we first present an overview of these methods with an emphasis on their computational efficiency. We then compare eight commonly used linear time complexity initialization methods on a large and diverse collection of data sets using various performance criteria. Finally, we analyze the experimental results using non-parametric statistical tests and provide recommendations for practitioners. We demonstrate that popular initialization methods often perform poorly and that there are in fact strong alternatives to these methods.

1. Introduction

Clustering, the unsupervised classification of patterns into groups, is one of the most important tasks in exploratory data analysis (Jain, Murty, & Flynn, 1999). Primary goals of clustering include gaining insight into data (detecting anomalies, identifying salient features, etc.), classifying data, and compressing data. Clustering has a long and rich history in a variety of scientific disciplines including anthropology, biology, medicine, psychology, statistics, mathematics, engineering, and computer science. As a result, a plethora of clustering algorithms have been proposed since the early 1950s (Jain, 2010).

Clustering algorithms can be broadly classified into two groups: hierarchical and partitional (Jain, 2010). Hierarchical algorithms recursively find nested clusters either in a top-down (divisive) or bottom-up (agglomerative) fashion. In contrast, partitional algorithms find all the clusters simultaneously as a partition of the data and do not impose a hierarchical structure. Most hierarchical algorithms have quadratic or higher complexity in the number of data points (Jain et al., 1999) and therefore are not suitable for large data sets, whereas partitional algorithms often have lower complexity.

Given a data set $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ in $\mathbb{R}^D$, i.e. N points (vectors) each with D attributes (components), hard partitional algorithms divide $\mathcal{X}$ into K exhaustive and mutually exclusive clusters $\mathcal{P} = \{P_1, P_2, \ldots, P_K\}$, $\bigcup_{i=1}^{K} P_i = \mathcal{X}$, $P_i \cap P_j = \emptyset$ for $1 \leqslant i \neq j \leqslant K$. These algorithms usually generate clusters by optimizing a criterion function. The most intuitive and frequently used criterion function is the Sum of Squared Error (SSE) given by:

$$\mathtt{SSE} = \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in P_i} \|\mathbf{x}_j - \mathbf{c}_i\|_2^2 \tag{1}$$

where $\|\cdot\|_2$ denotes the Euclidean ($\mathcal{L}_2$) norm and $\mathbf{c}_i = 1/|P_i| \sum_{\mathbf{x}_j \in P_i} \mathbf{x}_j$ is the centroid of cluster P_i whose cardinality is $|P_i|$. The optimization of (1) is often referred to as the minimum SSE clustering (MSSC) problem.

The number of ways in which a set of N objects can be partitioned into K non-empty groups is given by Stirling numbers of the second kind:

$$\mathcal{S}(N,K) = \frac{1}{K!} \sum_{i=0}^{K} (-1)^{K-i} \binom{K}{i} i^N \tag{2}$$

which can be approximated by $K^N/K!$ It can be seen that a complete enumeration of all possible clusterings to determine the global minimum of (1) is clearly computationally prohibitive except for very small data sets (Kaufman & Rousseeuw, 1990). In fact, this non-convex optimization problem is proven to be NP-hard even for $K = 2$ (Aloise, Deshpande, Hansen, & Popat, 2009) or $D = 2$ (Mahajan, Nimbhorkar, & Varadarajan, 2012). Consequently, various heuristics have been developed to provide approximate solutions to this problem (Tarsitano, 2003). Among these heuristics, Lloyd's algorithm (Lloyd, 1982), often referred to as the (batch) k-means algorithm, is the simplest and most commonly used one. This algorithm starts with K arbitrary centers, typically chosen uniformly at random from

[*] Corresponding author.
E-mail addresses: ecelebi@lsus.edu (M.E. Celebi), kingravi@gatech.edu (H.A. Kingravi), pvela@gatech.edu (P.A. Vela).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.07.021

the data points. Each point is assigned to the nearest center and then each center is recalculated as the mean of all points assigned to it. These two steps are repeated until a predefined termination criterion is met.

The k-means algorithm is undoubtedly the most widely used partitional clustering algorithm (Jain et al., 1999; Jain, 2010). Its popularity can be attributed to several reasons. First, it is conceptually simple and easy to implement. Virtually every data mining software includes an implementation of it. Second, it is versatile, i.e. almost every aspect of the algorithm (initialization, distance function, termination criterion, etc.) can be modified. This is evidenced by hundreds of publications over the last fifty years that extend k-means in various ways. Third, it has a time complexity that is linear in N, D, and K (in general, $D \ll N$ and $K \ll N$). For this reason, it can be used to initialize more expensive clustering algorithms such as expectation maximization (Bradley & Fayyad, 1998), DBSCAN (Dash, Liu, & Xu, 2001), and spectral clustering (Chen, Song, Bai, Lin, & Chang, 2011). Furthermore, numerous sequential (Kanungo et al., 2002; Hamerly, 2010) and parallel (Chen & Chien, 2010) acceleration techniques are available in the literature. Fourth, it has a storage complexity that is linear in N, D, and K. In addition, there exist disk-based variants that do not require all points to be stored in memory (Ordonez & Omiecinski, 2004). Fifth, it is guaranteed to converge (Selim & Ismail, 1984) at a quadratic rate (Bottou & Bengio, 1995). Finally, it is invariant to data ordering, i.e. random shufflings of the data points.

On the other hand, k-means has several significant disadvantages. First, it can only detect compact, hyperspherical clusters that are well separated. This can be alleviated by using a more general distance function such as the Mahalanobis distance, which permits the detection of hyperellipsoidal clusters (Mao & Jain, 1996). Second, due its utilization of the squared Euclidean distance, it is sensitive to noise and outlier points since even a few such points can significantly influence the means of their respective clusters. This can addressed by outlier pruning (Zhang & Leung, 2003) or using a more robust distance function such as City-block ($\mathcal{L}_1$) distance. Third, due to its gradient descent nature, it often converges to a local minimum of the criterion function (Selim & Ismail, 1984). For the same reason, it is highly sensitive to the selection of the initial centers. Adverse effects of improper initialization include empty clusters, slower convergence, and a higher chance of getting stuck in bad local minima (Celebi, 2011). Fortunately, all of these drawbacks except for the first one can be remedied by using an adaptive initialization method (IM).

In this study, we investigate some of the most popular IMs developed for the k-means algorithm. Our motivation is threefold. First, a large number of IMs have been proposed in the literature and thus a systematic study that reviews and compares these methods is desirable. Second, these IMs can be used to initialize other partitional clustering algorithms such as fuzzy c-means and its variants and expectation maximization. Third, most of these IMs can be used independently of k-means as standalone clustering algorithms.

This study differs from earlier studies of a similar nature (Pena, Lozano, & Larranaga, 1999; He, Lan, Tan, Sung, & Low, 2004) in several respects: (i) a more comprehensive overview of the existing IMs is provided, (ii) the experiments involve a larger set of methods and a significantly more diverse collection of data sets, (iii) in addition to clustering effectiveness, computational efficiency is used as a performance criterion, and (iv) the experimental results are analyzed more thoroughly using non-parametric statistical tests.

The rest of the paper is organized as follows. Section 2 presents a survey of k-means IMs. Section 3 describes the experimental setup. Section 4 presents the experimental results, while Section 5 gives the conclusions.

2. Initialization methods for k-means

In this section, we briefly review some of the commonly used IMs with an emphasis on their time complexity (with respect to N). In each complexity class, methods are presented in chronologically ascending order.

2.1. Linear time-complexity initialization methods

Forgy's method (Forgy, 1965) assigns each point to one of the K clusters uniformly at random. The centers are then given by the centroids of these initial clusters. This method has no theoretical basis, as such random clusters have no internal homogeneity (Anderberg, 1973).

Jancey's method (Jancey, 1966) assigns to each center a synthetic point randomly generated within the data space. Unless the data set fills the space, some of these centers may be quite distant from any of the points (Anderberg, 1973), which might lead to the formation of empty clusters.

MacQueen (1967) proposed two different methods. The first one, which is the default option in the Quick Cluster procedure of IBM SPSS Statistics (Norušis, 2011), takes the first K points in $\mathcal{X}$ as the centers. An obvious drawback of this method is its sensitivity to data ordering. The second method chooses the centers randomly from the data points. The rationale behind this method is that random selection is likely to pick points from dense regions, i.e. points that are good candidates to be centers. However, there is no mechanism to avoid choosing outliers or points that are too close to each other (Anderberg, 1973). Multiple runs of this method is the standard way of initializing k-means (Bradley & Fayyad, 1998). It should be noted that this second method is often mistakenly attributed to Forgy (1965).

Ball and Hall's method (Ball & Hall, 1967) takes the centroid of $\mathcal{X}$, i.e. $\overline{\mathcal{X}} = 1/N\sum_{j=1}^{N}\mathbf{x}_j$, as the first center. It then traverses the points in arbitrary order and takes a point as a center if it is at least T units apart from the previously selected centers until K centers are obtained. The purpose of the distance threshold T is to ensure that the seed points are well separated. However, it is difficult to decide on an appropriate value for T. In addition, the method is sensitive to data ordering.

The Simple Cluster Seeking method (Tou & Gonzales, 1974) is identical to Ball and Hall's method with the exception that the first point in $\mathcal{X}$ is taken as the first center. This method is used in the FASTCLUS procedure of SAS (SAS Institute Inc., 2009).

Späth's method (Späth, 1977) is similar to Forgy's method with the exception that the points are assigned to the clusters in a cyclical fashion, i.e. the j-th ($j \in \{1,2,\ldots,N\}$) point is assigned to the $(j - 1 \bmod K + 1)$-th cluster. In contrast to Forgy's method, this method is sensitive to data ordering.

Maximin method (Gonzalez, 1985; Katsavounidis, Kuo, & Zhang, 1994) chooses the first center $\mathbf{c}_1$ arbitrarily and the i-th ($i \in \{2,3,\ldots,K\}$) center $\mathbf{c}_i$ is chosen to be the point that has the greatest minimum-distance to the previously selected centers, i.e. $\mathbf{c}_1,\mathbf{c}_2,\ldots,\mathbf{c}_{i-1}$. This method was originally developed as a 2-approximation to the K-center clustering problem.[1] It should be noted that, motivated by a vector quantization application, Katsavounidis et al.'s variant (Katsavounidis et al., 1994) takes the point with the greatest Euclidean norm as the first center.

Al-Daoud's density-based method (Al-Daoud & Roberts, 1996) first uniformly partitions the data space into M disjoint hypercubes. It then randomly selects $K\,N_m/N$ points from hypercube m ($m \in \{1,2,\ldots,M\}$) to obtain a total of K centers (N_m is the number

[1] Given a set of N points in a metric space, the goal of K-center clustering is to find K representative points (centers) such that the maximum distance of a point to a center is minimized.