Expert Systems with Applications 42 (2015) 7541–7548

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Fuzzy C-means++: Fuzzy C-means with effective seeding initialization

Adrian Stetco *, Xiao-Jun Zeng, John Keane

School of Computer Science, University of Manchester, United Kingdom

ARTICLE INFO

Article history:
Available online 22 May 2015

Keywords:
Cluster analysis
Fuzzy C-means clustering
Initialization

ABSTRACT

Fuzzy C-means has been utilized successfully in a wide range of applications, extending the clustering capability of the K-means to datasets that are uncertain, vague and otherwise hard to cluster. This paper introduces the *Fuzzy C-means++* algorithm which, by utilizing the seeding mechanism of the K-means++ algorithm, improves the effectiveness and speed of Fuzzy C-means. By careful seeding that disperses the initial cluster centers through the data space, the resulting Fuzzy C-means++ approach samples starting cluster representatives during the initialization phase. The cluster representatives are well spread in the input space, resulting in both faster convergence times and higher quality solutions. Implementations in *R* of standard Fuzzy C-means and Fuzzy C-means++ are evaluated on various data sets. We investigate the cluster quality and iteration count as we vary the spreading factor on a series of synthetic data sets. We run the algorithm on real world data sets and to account for the non-determinism inherent in these algorithms we record multiple runs while choosing different k parameter values. The results show that the proposed method gives significant improvement in convergence times (the number of iterations) of up to 40 (2.1 on average) times the standard on synthetic datasets and, in general, an associated lower cost function value and Xie–Beni value. A proof sketch of the logarithmically bounded expected cost function value is given.

1. Introduction

Partitional cluster analysis is defined as the problem of partitioning a group of objects into clusters that share similar characteristics. The most well-known and widely used partitional clustering algorithms are K-means and Fuzzy C-means (Peizhuang, 1983). When compared across clusters, members of a cluster will be different from members of all other clusters. In order to quantify the similarity/dissimilarity relationship between objects, metric functions, defined on both numeric (Euclidean, Manhattan, Cosine, etc.) or non-numeric (Hamming, Jaro-Winkler, Levenshtein, etc.) data have been used.

K-means is one of the oldest clustering algorithms (MacQueen, 1967) and refers both to the clustering task and a specific algorithm to solve it. Given a set X of input data and a parameter k, the task is to choose k representatives of X such that the distances between any points in X and their representative is minimized. The set of representatives discovered after running the K-means algorithm is enough to define a clustering of the points in the data space (the ith cluster being the set of all points in X that are closer to r_i than any other representative).

In contrast to the of K-means where each point belongs to one cluster, in Fuzzy C-means each point x_i in the space belongs to r_j, $\forall j \in R$ with $\mu_{ij} \in [0,1]$ defined in the membership matrix (of size $n \times k$ where n is the number of points in the data space and k is the number of representatives). The use of a membership matrix increases the expressiveness of the clustering analysis, arguably presenting a more comprehensive view of relationships present in the data. Further, the hard assignment of the data points by K-means is inadequate when the points are equally distanced between representatives, in which case they will be randomly assigned to one cluster or another (Doring, Lesot, & Kruse, 2006). Fuzzy C-means mitigates this problem by assigning equal degrees of belonging through the use of the membership matrix. This method computes membership degrees at each iteration, a costly operation that gives a membership degree to a point proportional to its proximity to the cluster representatives. Moreover, the size of this matrix grows as a product of the number of points and clusters, making the algorithm computationally expensive for high values. To reduce the computational burden of the algorithm and at the same time increase its accuracy, an integration of the K-means careful seeding algorithm (Arthur, Arthur, Vassilvitskii, & Vassilvitskii, 2007) into the standard version of Fuzzy C-means is proposed, analyzed and verified in this paper.

The reminder of the paper is structured as follows: Section 2 presents work improving the performance of Fuzzy C-means; both

* Corresponding author.
E-mail addresses: stetcom@cs.man.ac.uk (A. Stetco), x.zeng@manchester.ac.uk (X.-J. Zeng), john.keane@manchester.ac.uk (J. Keane).

http://dx.doi.org/10.1016/j.eswa.2015.05.014
0957-4174/

the standard and the proposed algorithm are introduced in Section 3, together with a proof that shows the theoretical bounds of the expected cost function; Section 4 presents the datasets and the evaluation procedure used, and compares the proposed scheme with the standard algorithm; Section 5 summarizes findings and considers future work.

2. Background

Although noted both for its simplicity of implementation and its output validity, Fuzzy C-means suffers from high computational cost. For each iteration the computational complexity of the algorithm is quadratic in the number of clusters $O(NC\hat{}2P)$ where N is the number of data points, C is the number of clusters and P is the dimension of the data points. A linear complexity approach $O(NCP)$ that removes the need to store a large matrix during the iterations was proposed in Kolen and Hutcheson (2002). In Wang, Wang, and Wang (2004) a method to obtain qualitatively better clusters (as measured using a series of validity indexes) is proposed. This approach uses a weighted Euclidean distance which incorporates feature weights. While this method showed promising results on several UCI databases, it requires a feature weight learning step of complexity $O(N^{2CP})$.

Work by Zou, Wang, and Hu (2008) addresses the problem of initializing the cluster representatives by partitioning the space into grid blocks (finite disjoint rectangle-like units) and performing a search for condensation points. A grid block is considered dense if the number of data points present in it are bigger than a given input threshold parameter. Condensation points are geometric centers of dense grid blocks and serve as good initialization points to be chosen as cluster center before commencing the Fuzzy C-means algorithm. Although this method works well on two-dimensional datasets, the question remains how well it would work for non-spherical cluster types, and what should the block sizes and density threshold values be.

Yang, Zhang, and Tian (2010) propose a methodology for picking centres based on subtractive clustering. The potential of each point to become an initial centre is a function of its neighboring points: the more neighbors the higher the chance of being picked. Although promising, being able to select the number of k parameter as well as initializing the algorithm, this method lacks enough empirical tests on real world datasets. Moreover it has four additional parameters that need tuning.

Celebi, Kingravi, and Vela (2013) conducted a comparative study on eight linear-time initialization techniques for K-means algorithm on a large variety of data sets. The study has looked at the quality (taking into consideration cost function values, external validity index) and speed number of iterations and CPU time) of the approaches. While most of these methods were non-deterministic (generating different initial points), two of them were deterministic, picking the same starting points every time when executed on the same data set. On real data sets, the non-deterministic methods (such as K-means++ (Arthur et al., 2007)) performed better than deterministic ones (with respect to minimum statistic), a fact that can be attributed to multiple local minima of the datasets and the fact that they were executed multiple times. However, the authors' argue that deterministic methods need only one run, thus total computational complexity could be lower in their case.

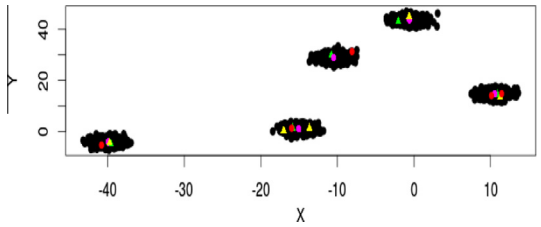

Fig. 1. Equal sized clusters with no overlap. Real cluster centers are marked in magenta, Fuzzy C-means initial clusters are marked in red, while in the yellow and green triangles we have Fuzzy C-means++ (with $p = 0.5$ and $p = 1.8$ respectively). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The K-means++ method (Arthur et al., 2007), the basis of this work, initializes the cluster centers of the K-means algorithm by selecting points in the dataset that are further away from each other in a probabilistic manner. This method both avoids the problems of the standard method and improves speed of convergence, being theoretically guaranteed to be $O(\log k)$, and hence competitive with the optimal solution. While Celebi et al. (2013) used the standard K-means++ initialization method in their study, we focus on the more general case and apply it to Fuzzy C-means, using a parameter to control the spreading. This method improves the way in which Fuzzy C-means initializes its clusters and has several advantages over the methods discussed. The method achieves superior clustering (in terms of validity indexes) compared to using a random initialization as in the standard and fewer iterations. The proposed method is also easier to understand and implement and compared to other methods it needs just one parameter that controls the spreading factor.

The R programming language (R Development Core Team, 2013) is used here with the e1071 package (Meyer, Dimitriadou, Hornik, Weingessel, & Leisch, 2014) which, as well as containing standard clustering algorithms, contains useful cluster validity functions to test the quality of the discovered structures.

3. The algorithms

3.1. Fuzzy C-means algorithm

The standard version of the Fuzzy C-means algorithm (Peizhuang, 1983) – Algorithm 1) – minimizes the function:

$$\boldsymbol{J_m}(\boldsymbol{U},\boldsymbol{R}) = \sum_{i=1}^{N}\sum_{j=1}^{K}\mu_{ij}^{m}|x_i - r_j|^2 \tag{1}$$

subject to

$$\mu_{ij} \in [0,1]; \quad \sum_{j=1}^{k}\mu_{ij} = 1\forall i; \quad 0 < \sum_{i=1}^{n}\mu_{ij} < N, \quad \forall N$$

Algorithm 1: Fuzzy C-means (FCM)
Given X= $\{\boldsymbol{x_i}\}_{i=1}^{N}$ **and** $\boldsymbol{k}$**, return U and R**
1: procedure FCM (Data set X, Clusters k)
2: U^0 is randomly initialized
3: repeat
4: $r_j = \frac{1}{\sum_{i=1}^{n}\mu_{ij}^{m}}\sum_{i=1}^{n}\mu_{ij}^{m}x_i, \quad j = 1\ldots k$
5: $u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{|x_i - r_j|}{|x_i - r_k|}\right)^{\frac{2}{m-1}}}$ (3)
6: until $|U^{k+1} - U^k| < e$
7: end procedure

where $\mathbf{X} = \{x_i\}_{i=1}^{N}$ the set of data points, $\mathbf{U} = \{\mu_{ij}\}_{i,j=1}^{NK}$ the matrix of membership degrees, $k \in N$ the number of clusters and $\mathbf{R} = \{r_i\}_{i=1}^{k}$ the set of representatives, m is the fuzzifier parameter which