Contents lists available at ScienceDirect

Pattern Recognition

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

Fast modified global *k*-means algorithm for incremental cluster construction

Adil M. Bagirov [*], Julien Ugon, Dean Webb

Centre for Informatics and Applied Optimization, Graduate School of Information Technology and Mathematical Sciences, University of Ballarat, Victoria 3353, Australia

ARTICLE INFO

Article history:
Received 23 February 2010
Received in revised form
24 September 2010
Accepted 24 October 2010

Keywords:
Minimum sum-of-squares clustering
Nonsmooth optimization
k-means algorithm
Global *k*-means algorithm

ABSTRACT

The *k*-means algorithm and its variations are known to be fast clustering algorithms. However, they are sensitive to the choice of starting points and are inefficient for solving clustering problems in large datasets. Recently, incremental approaches have been developed to resolve difficulties with the choice of starting points. The global *k*-means and the modified global *k*-means algorithms are based on such an approach. They iteratively add one cluster center at a time. Numerical experiments show that these algorithms considerably improve the *k*-means algorithm. However, they require storing the whole affinity matrix or computing this matrix at each iteration. This makes both algorithms time consuming and memory demanding for clustering even moderately large datasets. In this paper, a new version of the modified global *k*-means algorithm is proposed. We introduce an auxiliary cluster function to generate a set of starting points lying in different parts of the dataset. We exploit information gathered in previous iterations of the incremental algorithm to eliminate the need of computing or storing the whole affinity matrix and thereby to reduce computational effort and memory usage. Results of numerical experiments on six standard datasets demonstrate that the new algorithm is more efficient than the global and the modified global *k*-means algorithms.

1. Introduction

Cluster analysis, also known as unsupervised data classification, is an important subject in data mining. Its aim is to partition a collection of patterns into clusters of similar data points. There are different types of clustering and in this paper we consider the unconstrained hard clustering problem. The *k*-means algorithm and its variations are known to be fast algorithms for solving such problems. However, they are sensitive to the choice of starting points and can only be applied to small datasets.

One common way of avoiding this problem is to use the multi restarting *k*-means algorithm. However, as the size of the dataset and the number of clusters increase, more and more starting points are needed to get a near global solution to the clustering problem. Consequently the multi restarting *k*-means algorithm becomes very time consuming and inefficient for solving clustering problems, even in moderately large datasets [1].

Different approaches to improve the efficiency of the *k*-means algorithm have been proposed, of which incremental ones are among the most successful. In these approaches clusters are computed incrementally by solving all intermediate clustering problems [1–4]. The global *k*-means algorithm proposed in [4] and the modified global *k*-means algorithm proposed in [1,5] are incremental clustering algorithms. Results of numerical experiments presented in [1] show that these algorithms allow one to find global or a near global minimizer of the cluster (or error) function. However, these algorithms are memory demanding as they require the storage of the affinity matrix. Alternatively, this matrix can be computed at each iteration, however, this extends the computational time significantly.

In this paper, a new version of the modified global *k*-means algorithm is proposed. We apply an auxiliary cluster function, introduced in [1], to generate a set of starting points lying in different parts of the dataset. The *k*-means algorithm is applied starting from these points to minimize the auxiliary cluster function and the best solution is selected as a starting point for the next cluster center. We exploit information gathered in previous iterations of the incremental algorithm to avoid computing the whole affinity matrix. Also the triangle inequality for distances is used to avoid unnecessary computations. We present results of numerical experiments on six standard datasets. These results demonstrate that the proposed algorithm is far more efficient than the global and modified global *k*-means algorithms.

The proposed algorithm is applicable to datasets with only numeric attributes. Clustering algorithms for categorical datasets can be found, for example, in [6].

It should be noted that in [7] the authors propose a fast version of the modified global *k*-means algorithm. However, their aim is to reduce computational complexity, whereas the purpose of this paper is to reduce memory usage. Indeed, the algorithm proposed

[*] Corresponding author.
E-mail address: a.bagirov@ballarat.edu.au (A.M. Bagirov).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.10.018

in [7] still requires computing part of the affinity matrix, which we specifically avoid.

The rest of the paper is organized as follows. We give a brief overview of incremental clustering algorithms in Section 2. In Section 3, the modified global k-means algorithm is described. An algorithm for solving the auxiliary problem is discussed in Section 4. Section 5 presents approaches to reduce computational effort. The numerical results are given in Section 6. Section 7 concludes the paper.

2. Brief overview of incremental algorithms for clustering

Incremental approaches are becoming very popular in data mining in general and in cluster analysis in particular. The paper [8] is one of the first introducing the incremental algorithm COBWEB for conceptual clustering.

The existing incremental algorithms in cluster analysis can be divided, without any loss of generality, into the following classes:

1. Algorithms where new data points are added at each iteration and cluster centers are refined accordingly. Such algorithms are called *single pass incremental algorithms*;
2. Algorithms where clusters are built incrementally adding one cluster center at a time.

Single pass incremental algorithms are applicable to very large datasets. In recent years, there has been a dramatic growth of interest in developing such algorithms for massive datasets. In particular, clustering in the streamed datasets has received a lot of attention. Here, an algorithm must process its input by making one or a few passes over it, using a limited amount of memory. This is a common model when the size of the input data far exceeds the size of the memory available. Over last several years various approaches to design single pass algorithms have been proposed (see, for example, [9–12]).

The algorithms discussed in this paper belong to the second class. Unlike those from the first class, algorithms from this class compute clusters incrementally using the whole dataset and the number of passes is not restricted. These algorithms are not directly applicable to solve clustering problems in massive datasets. In order to solve k-partition clustering problem these algorithms start from one cluster center (centroid of the dataset) and compute cluster centers incrementally adding a new center at each iteration. The algorithms are capable of finding either global or near global solutions to clustering problems in many datasets. Only these solutions provide best cluster structure of the dataset. Different incremental algorithms have been proposed in [1–4]. Although these algorithms are robust and accurate they are time consuming as the computation of the affinity matrix is required at each iteration. The modified global k-means algorithm proposed in [1,5] is an incremental clustering algorithm. Results presented in these papers demonstrate that it can find better solutions than other incremental algorithms in many datasets. However, this algorithm uses significantly more computational effort than other incremental algorithms.

In this paper, we propose a new version of the modified global k-means algorithm. In this version we reduce the amount of computational effort by:

1. removing data points which are close to cluster centers found in the previous iteration. Thus we use the whole dataset only at the first iteration when we compute the centroid of the dataset;
2. using the triangle inequality for distances to avoid unnecessary computations.

We also introduce a scheme to generate starting points from different parts of the dataset to minimize the auxiliary function.

It should be noted that several approaches have been proposed to accelerate the k-means algorithm. Among them, the authors in [13] propose to collect information on the data in a tree so that nearby points are in the same subtree. In [14–16], authors use the triangle inequality for distances to avoid unnecessary calculations by reusing information collected in previous iterations of the k-means algorithm. The paper [7] propose the new version of the global k-means algorithm which is applicable to large datasets and uses the cluster membership and geometrical information of a data point. We use a similar approach, but we also reuse information collected while solving the problem with fewer centers. This allows us to solve simpler and simpler problems as we increment the number of clusters, by relying on the fact that adding clusters brings more structure to the result.

3. Modified global k-means algorithm

In this section we briefly describe the modified global k-means algorithm. The detailed description of this algorithm can be found in [1].

In cluster analysis we assume that we have been given a finite set of points A in the n-dimensional space $\mathbb{R}^n$, that is

$$A=\{a^1,\ldots,a^m\} \quad \text{where } a^i \in \mathbb{R}^n,\ i=1,\ldots,m.$$

We consider the hard unconstrained partition clustering problem, that is the distribution of the points of the set A into a given number k of disjoint subsets $A^j, j=1,\ldots,k$ with respect to predefined criteria such that

(1) $A^j \neq \emptyset, \quad j=1,\ldots,k$;
(2) $A^j \cap A^l = \emptyset, \quad j,l=1,\ldots,k,\ j\neq l$;
(3) $A=\bigcup_{j=1}^{k} A^j$;
(4) no constraints are imposed on the clusters $A^j, j=1,\ldots,k$.

The sets $A^j, j=1,\ldots,k$ are called clusters. We assume that each cluster A^j can be identified by its center (or centroid) $x^j \in \mathbb{R}^n, j=1,\ldots,k$. There are different reformulations of the clustering problem as an optimization problem. A nonsmooth, nonconvex optimization formulation is as follows (see [17–19]):

$$\text{minimize } f_k(x) \text{ subject to } x=(x^1,\ldots,x^k)\in\mathbb{R}^{n\times k}, \tag{1}$$

where

$$f_k(x^1,\ldots,x^k)=\frac{1}{m}\sum_{i=1}^{m}\min_{j=1,\ldots,k}\|x^j-a^i\|^2. \tag{2}$$

Here $\|\cdot\|$ is the Euclidean norm. In this case the problem (1) is also known as the minimum sum-of-squares clustering problem. Many algorithms have been developed to solve problem (1) (see, for example, [20–32]). Over the last several years different incremental algorithms have been also proposed to solve it (see [1–4]).

The modified global k-means algorithm, introduced in [1], is an incremental clustering algorithm. To solve k-partition problem, this algorithm starts with the computation of one cluster center, that is with the computation of the centroid of the dataset, and attempts to optimally add one new cluster center at each iteration. The k-partition problem is solved using the $k-1$ centers for the $(k-1)$-partition problem and the remaining k-th center is placed in an appropriate place. An auxiliary cluster function is defined using $k-1$ cluster centers from the $(k-1)$-th iteration and is minimized to compute the starting point for the k-th center. Then this new center together with previous $k-1$ cluster centers is taken as a starting point for the k-partition problem. The k-means algorithm is applied starting from this point to find the k-partition of the dataset. Such an approach allows one to find a global or a near global solution to problem (1).