Expert Systems with Applications 37 (2010) 5645–5652

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A new clustering algorithm based on hybrid global optimization based on a dynamical systems approach algorithm

Ali Maroosi *, Babak Amiri

Iran University of Science and Technology, Tehran, Iran

ARTICLE INFO

Keywords:
Clustering
K-means
Dynamical systems
Tabu search

ABSTRACT

Many methods for local optimization are based on the notion of a direction of a local descent at a given point. A local improvement of a point in hand can be made using this direction. As a rule, modern methods for global optimization do not use directions of global descent for global improvement of the point in hand. From this point of view, global optimization algorithm based on a dynamical systems approach (GOP) is an unusual method. Its structure is similar to that used in local optimization: a new iteration can be obtained as an improvement of the previous one along a certain direction. In contrast with local methods, is a direction of a global descent and for more diversification combined with Tabu search. This algorithm is called hybrid GOP (HGOP). Cluster analysis is one of the attractive data mining techniques that are used in many fields. One popular class of data clustering algorithms is the center based clustering algorithm. *K*-means is used as a popular clustering method due to its simplicity and high speed in clustering large datasets. However, *K*-means has two shortcomings: dependency on the initial state and convergence to local optima and global solutions of large problems cannot found with reasonable amount of computation effort. In order to overcome local optima problem lots of studies have been done in clustering. In this paper, we proposed application of hybrid global optimization algorithm based on a dynamical systems approach. We compared HGOP with other algorithms in clustering, such as GAK, SA, TS, and ACO, by implementing them on several simulation and real datasets. Our finding shows that the proposed algorithm works better than others.

1. Introduction

Clustering, so-called set partitioning, is a basic and widely applied methodology. Application fields include statistics, mathematical programming (such as location selecting, network partitioning, routing, scheduling and assignment problems, etc.) and computer science (including pattern recognition, learning theory, image processing and computer graphics, etc.). Clustering is mainly to group all objects into several mutually exclusive clusters in order to achieve the maximum or minimum of an objective function. Clustering is rapidly becoming computationally intractable as problem scale increases, because of the combinatorial character of the method. Brucker (1978) and Ward (1963) proved that, for specific object functions, clustering becomes an NP-hard problem when the number of clusters exceeds 3.

There are many methods applied in clustering analysis, like hierarchical clustering, partition-based clustering, density-based clustering, and artificial intelligence-based clustering.

One popular class of data clustering algorithms is the center based clustering algorithm. *K*-means is used as a popular clustering method due to its simplicity and high speed in clustering large datasets (Forgy, 1965). However, *K*-means has two shortcomings: dependency on the initial state and convergence to local optima (Selim & Ismail, 1984) and also global solutions of large problems cannot found with reasonable amount of computation effort (Spath, 1989). In order to overcome local optima problem lots of studies have been done in clustering.

Mualik and Bandyopadhyay (2000) proposed a genetic algorithm based method to solve the clustering problem and experiment on synthetic and real life datasets to evaluate the performance. The results showed that GA-based method might improve the final output of *K*-means.

Krishna and Murty (1999) proposed a novel approach called genetic *K*-means algorithm for clustering analysis. It defines a basic mutation operator specific to clustering called distance-based mutation. Using finite Markov chain theory, it proved that GKA converge to the best-known optimum.

Selim and Al-Sultan (1991) discussed the solution of the clustering problem usually solved by the *K*-means algorithm. The is problem known to have local minimum solutions, which are

* Corresponding author.
E-mail addresses: Ali.Maroosi@gmail.com, Ali_maroosi@ee.iust.ac.ir (A. Maroosi), Amiri_babak@ind.iust.ac.ir (B. Amiri).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.02.047

usually what the *K*-means algorithm obtains. The simulated annealing approach for solving optimization problems described and proposed for solving the clustering problem. The parameters of the algorithm were discussed in detail and it was shown that the algorithm converges to a global solution of the clustering problem.

According to Sung and Jin (2000), researchers considered a clustering problem where a given data set partitioned into a certain number of natural and homogeneous subsets such that each subset is composed of elements similar to one another but different from those of any other subset. For the clustering problem, a heuristic algorithm exploited by combining the Tabu search heuristic with two complementary functional procedures, called packing and releasing procedures. The algorithm was numerically tested for its electiveness in comparison with reference works including the Tabu search algorithm, the *K*-means algorithm and the simulated annealing algorithm.

Over the last decade, modeling the behavior of social insects, such as ants and bees, for the purpose of search and problem solving has been the context of the emerging area of swarm intelligence. Using ant colony is a typical successful swarm-based optimization approach, where the search algorithm is inspired by the behavior of real ants.

Kuo, Wang, Hu, and Chou (2005) proposed a novel clustering method, ant *K*-means (AK) algorithm. Ant *K*-means algorithm modifies the *K*-means as locating the objects in a cluster with the probability, which updated by the pheromone, while the rule of updating pheromone is according to total within-cluster variance (TWCV).

Shelokar, Jayaraman, and Kulkarni (2004) presents an ant colony optimization, methodology for optimally clustering N objects into K clusters. The algorithm employs distributed agents who mimic the way real ants find a shortest path from their nest to food source and back. They compared result with other algorithms in clustering, GA, Tabu search, SA. They showed that their algorithms are better than other algorithms in performance and time.

This paper presents application of HGOP algorithm for clustering. The paper is organized as follows: in Section 2 we discussed cluster analysis problems. Section 3 introduces HGOP philosophy and application of it on clustering, and then in Section 4 experimental result of proposed clustering algorithm in comparison with other clustering algorithms is shown.

2. Clustering

Data clustering, which is an NP-complete problem of finding groups in heterogeneous data by minimizing some measure of dissimilarity, is one of the fundamental tools in data mining, machine learning and pattern classification solutions (Garey, Johnson, & Witsenhausen, 1982). Clustering in N-dimensional Euclidean space RN is the process of partitioning a given set of *n* points into a number, say *k*, of groups (or, clusters) based on some similarity (distance) metric in clustering procedure which is Euclidean distance, derived from the Minkowski metric (Eqs. (1) and (2)).

$$\mathrm{d}(x,y)=\left(\sum_{i=1}^{m}|x_i-y_j|^{r}\right)^{1/r} \tag{1}$$

$$\mathrm{d}(x,y)=\sqrt{\sum_{i=1}^{m}(x_i-y_j)}^{2} \tag{2}$$

Let the set of *n* points $\{X_1,X_2,\ldots,X_n\}$ be represented by the set *S* and the *k* clusters be represented by $C_1,C_2\ldots,C_K$. Then:

$$\begin{aligned}&C_i\neq\phi \quad \text{for } i=1,\ldots,k,\\&C_i\cap C_j=\phi \quad \text{for } i=1,\ldots,k, j=1,\ldots,k, \text{ and } i\neq j\\&\text{and } \bigcup_{i=1}^{K}C_i=S\end{aligned}$$

In this study, we will also use Euclidian metric as a distance metric. The existing clustering algorithms can be simply classified into the following two categories: hierarchical clustering and partitional clustering. The most popular class of partitional clustering methods are the center based clustering algorithms (Gungor & Unler, 2006). The *K*-means algorithms, is one of the most widely used center based clustering algorithms (Forgy, 1965). To find *K* centers, the problem is defined as an optimization (minimization) of a performance function, $f(X,Z)$, defined on both the data items and the center locations. A popular performance function for measuring goodness of the k clustering is the total within-cluster variance or the total mean-square quantization error (MSE), Eq. (3) (Gungor & Unler, 2006).

$$f(X,Z)=\sum_{i=1}^{N}Min\{\|X_i-Z_l\|^2| \quad l=1,\ldots,K.\} \tag{3}$$

The steps of the *K*-means algorithm are as follow (Mualik & Bandyopadhyay, 2000):

Step 1: Choose *K* cluster centers $Z_1,\ Z_2,\ldots,Z_k$ randomly from *n* points $\{X_1,X_2,\ldots,X_n\}$.
Step 2: Assign point $X_i,\ i=1,2,\ldots,n$ to cluster $C_j,\ J\in\{1,2,\ldots,K\}$ if $\|X_i-Z_j\|<\|X_i-Z_p\|$, *p*=1, 2, …, *K*, and $j\neq p$.
Step 3: Compute new cluster centers $Z_1^*,Z_2^*,\ldots,Z_K^*$ as follows:

$$Z_i^*=\frac{1}{n}\sum_{x_j\in C_i}X_j, \quad i=1,2,\ldots,K,$$

where n_i is the number of elements belonging to cluster C_i.
Step 4: If termination criteria are satisfied, stop otherwise continues from step 2

Note that in case the process does not terminate at step 4 normally, then it is executed for a mutation fixed number of iterations.

Global optimization algorithm (GOP)

Steps of the global optimization algorithm are as follow:

1. Select some initial random points that is define a set $A=\{U(t)=(U1(t),\ldots,Un(t));\ t=1,\ldots,T\}$; the set A uniformly select from the box $U\in R^{Kd}, a_i\leqslant u_i\leqslant b_i,\ i=1,\ldots,Kd$;.
2. Calculate the performance value at set A and then chose a point $U^*\in A$which provides the best performance value and $U_T=U^*$.
3. Find good point U_{T+1} from $U^*(U_T)$ and add it to the set A. For each point $U\in A$ and each coordinate i calculate degree of the change of objective function values *f*(U) when u_i changes. Here change means either decrease or increase of a scalar variable.
4. $U_T=U^*$ that $U^*=(u_0^*,u_1^*,\ldots,u_n^*)$, calculate $F(u_i\uparrow)-F(u_i\downarrow)$.
5. Using these degrees, calculate forces $F(T)=(F_1(T),\ldots,F_{KL}(T))$ acting on increase of objective function values *f* at the point $U^*(U_T)$. $F(T)=F(u_i\uparrow)-F(u_i\downarrow)$.
6. Calculate U_{T+1} from U_T by $U_{T+1}=U_T+\alpha F(T)$.
7. Then by the same manner, choose a new point U_{T+2} and so on.
8. This process is terminated if either $F(T)=0$ or after maximum iteration hold the best solution and repeat all of the these procedures with take other initial random points and another stage again.