Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online
CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-12 (2016) 243–248

BAT-CLARA: BAT-inspired algorithm for Clustering LARge Applications

Yasmine Aboubi, Habiba Drias, Nadjet Kamel

LRIA , USTHB,
USTHB, BP 32 El Alia, Bab Ezzouar Algiers, Algeria
(e-mail: yasminaboubi@gmail.com, h_drias@hotmail.fr, nadjet.kamel@gmail.com)

www.lria.usthb.dz

Abstract: Bat algorithm is a new nature-inspired metaheuristic optimization algorithm introduced by Yang in 2010, especially based on echolocation behavior of microbats when searching their prey. Firstly, this algorithm is used to solve various continuous optimization problems. Clustering remains one of the most difficult challenges in data mining. In this paper, an overview of literature methods is undertaken followed by the presentation of a new algorithm called BAT-CLARA for clustering large data sets. It is based on bat behavior and k-medoids partitioning. The new technique is compared to the well-know partitioning algorithms PAM, CLARA, CLARANS and CLAM, a recent algorithm found in the literature. Experimental results show that, for the same tested datasets, BAT-CLARA is more effective and more efficient than previous clustering methods.

Keywords: clustering algorithms, bat-inspired algorithm, metaheuristics, medoids

1. INTRODUCTION

Data clustering is a fundamental problem in a variety of areas of computer science and related fields, such as datamining, data compression, statistical data analysis. It aims at gathering data into groups regarding to their similarity. One important target for such datamining task is to reduce the data size and hence ensures scalability. A cluster is a collection of objects which are similar between themselves and are dissimilar to the objects belonging to the other clusters. In other words, the goal of clustering is to distribute data into clusters such that the similarities among objects within the same cluster are maximal while similarities among objects from different cluster are minimal.

Clustering algorithms are generally classified as hierarchical and partitionning algorithms. In this paper, we are interested in partitioning algorithms, as they are widely used, Often the k clusters found by a partitioning method are of higher quality than the k clusters produced by hierarchical method (Raymond T and Han, 2002). The problem of partitioning n objects over k clusters is important as it has numerous applications in diverse domains. It is NP-hard and regarding this fact, a lot of efforts have been devoted to its resolution. As evoked previously, the objective is to put together the objects having similar characteristics in one cluster. We need therefore to define a similarity measure between the objects. An effective partitioning is such that the similarity between objects of the same cluster is maximal and the similarity between objects of different clusters is minimal. In order to facilitate these measures, clusters are represented by objects, generally expressed by some statistics such as means or medians. Concretely, from the optimization viewpoint, the issue consists in determining clusters such that to minimize the sum of the dissimilarities between each object and the representative of the cluster to which it belongs. This quantity, expressed by Formula (1), represents the absolute error, also called inertia of the partitioning. I is the sum of the absolute error for all objects in the data set, p is a given object in cluster C_j and o_j is the center or representative object of C_j, k being the number of clusters.

$$I = \sum_{j=1}^{k} \sum_{p \in C_j} d(p, o_j) \tag{1}$$

The data mining community has recently deployed a lot of efforts on developing fast algorithms for partitioning very large data sets. An overview of existing methods was elaborated in order to compare our proposal to the previous techniques. The most popular clustering algorithms are k-means (MacQueen (1967)), PAM (Leonard and Peter (1990)), CLARA (Leonard and Peter (1990)) and CLARANS (Raymond T and Han (2002)). CLAM (Nguyen, Quynh H. and Rayward-Smith, Victor J. (2011)) is a recent original approach exploring metaheuristcs for clustering.

Following the last mentioned research direction, we propose a new clustering algorithm based on Bat-inspired computing for Clustering LARge Application, namely Bat-CLARA. As its name suggests, the algorithm uses the Bat optimization technique to explore the solutions space in an efficient way. It also uses some necessary concepts of PAM to build clusters. Extensive experiments were performed on real datasets to validate the effectiveness and efficiency of the proposed algorithm relatively to the existing techniques of the literature.

This paper is structured as follows. Section 2 presents

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.607

related works used the most by the scientific community as well as recent ones. Section 3 describes the metaheuristic inspired by Bats behavior and based on collective intelligence. The new clustering algorithm called Bat-CLARA is introduced in section 4. whilst section 5 provides the results of the experiments that were conducted for the validation of the proposal.

2. RELATED WORKS

The most known clustering techniques are k-means, k-medoids, CLARA and CLARANS. In this section, we describe their respective method as well as their benefits and disadvantages.

2.1 k-means

k-means is the most widely used clustering algorithm because of the simplicity of its implementation. It starts by drawing at random k centers then it assigns each objet to a cluster according to its distance with the cluster center. The means is calculated for each cluster once all the objects are inserted in the clusters and becomes the new center. This process is repeated until no changes occur in the centers. This algorithm is known to be not effective enough because of the representation of the center as the means of the objects residing inside the cluster. However it is efficient as it consists of one loop inside which, we dispatch the objects over the clusters and calculate the means with a simple formula.

2.2 PAM

k-medoids also called PAM (Partition Around Medoids) (Leonard and Peter (1990)) was designed to palliate to the k-means lack in effectiveness. It shares the same algorithmic structure but uses medoids as cluster representatives. The fact to substitute the means by the medoid makes the algorithm more effective because the medoid position in the cluster is central and the error value of Formula (1) decreases. Another benefit is the insensitivity to noise and outlier. However, it is less efficient.

2.3 CLARA

CLARA stands for **Cl**ustering **LAR**ge **A**pplication (Leonard and Peter (1990)). It is an improvement of PAM to handle large datasets. It runs PAM on multiple random samples, instead of the whole dataset. Experiments reported in (Leonard and Peter (1990)) indicate that five samples of size $(40+2k)$ give satisfactory results. It has been shown to produce relatively good quality solutions in a reasonable computation time for large data sets but it is less effective as it considers samples and not the entire datasets.

2.4 CLARANS

CLARANS (Ng and Han 1994) is proposed for **CL**ustering Large **A**pplications with **RAN**domized **S**earch in order to improve the effectiveness in comparison to CLARA. The algorithm uses a sampling technique to reduce the search space and the sampling is conducted dynamically for each iteration of the search procedure.
Conceptually, the clustering process can be viewed as a search through a graph $G_{n,k}$, where each vertex is a collection of k medoids $O_{m1}, ., O_{mk}$. Two nodes $S_1 = O_{m1}, , O_{mk}$ and $S_2 = O_{w1}, ..., O_{wk}$ are neighbors (that is, connected by an eadge in the graph) if their sets differ by only one object $|S1 \cap S2| = k - 1$. Each node can be assigned a cost that is defined by the total dissimilarity between every object and the medoids of its cluster. At each step, PAM examines all of the neighbors of the current node in its search for a minimum cost solution. As a consequence, the dynamic sampling used in CLARANS is more effective than the method used in CLARA in large application and more efficient than the swap phase of PAM in small application (Raymond T and Han (2002)). It outperforms then all the previous partitioning methods described previously.

2.5 CLAM

CLAM (Nguyen, Quynh H. and Rayward-Smith, Victor J. (2011)) stands for **C**lustering **L**arge **A**pplication using **M**etaheuristic). This algorithm uses a hybrid metaheuristic, combining Variable Neighborhood Search (VNS) and Tabu Search to guide the search to solve the problem of k-medoids clustering. Compared to CLARANS, experimental results show that, given the same computation times, CLAM is more effective. The only case where CLARANS outperforms CLAM is when both algorithms are set to perform a very small number of moves in the search space. In CLAM the search space is represented by a graph $G_{n,k}$, since each node has $k(n-k)$ neghboors. CLAM started with a randomly node in $G_{n,k}$; this search space is highly inter-connected, the number of shared neighbours between two neighbouring solutions is $(n2)$.
The revers VNS represents an intensification in the search strategy, the adoption of tabu list enforces the diversification of the search, CLAM employs a fixed a length of tabu liste firstin-first-out. Before a solution is considered, it is first checked against the tabu list. If the solution is not in the list, it is then visited and added to the list. If the solution is already in the list, it is discarded and the search can move on to another solution.

The first step in CLAM is to initialize a suitable value for the diameter (dim) of the datasets, defined as the largest distance between any tow data objects, so the suitable value for the radius range $[star\_radius, end\_radius]$ and the number of steps (t). For example a setting might be (50%, 40%, 30%, 20%, 10%) of dim, giving $start\_radius = dim/2$, $end\_radius = dim/10$ and $t = 5$. and $radius\_index = 1, 2, 3, 4, ...t$ as a pointer to the actual radius.

CLAM is described in Algorithm 1

Algorithm 1. CLAM

```
Input:
   k the number of clusters;
   D a set of n objects;
Output:
   a set of k clusters;
begin
```