An Efficient *K*-Medoids-Based Algorithm Using Previous Medoid Index, Triangular Inequality Elimination Criteria, and Partial Distance Search

Shu-Chuan Chu[1], John F. Roddick[1], and J.S. Pan[2]

[1] School of Informatics and Engineering,
Flinders University of South Australia,
PO Box 2100, Adelaide 5001, South Australia.
{jan, roddick}@cs.flinders.edu.au
[2] Department of Electronic Engineering,
Kaohsiung University of Applied Sciences
415 Chien Kung Road,
Kaohsiung, Taiwan
jspan@cc.kuas.edu.tw

Abstract. Clustering in data mining is a discovery process that groups similar objects into the same cluster. Various clustering algorithms have been designed to fit various requirements and constraints of application. In this paper, we study several k-medoids-based algorithms including the PAM, $CLARA$ and $CLARANS$ algorithms. A novel and efficient approach is proposed to reduce the computational complexity of such k-medoids-based algorithms by using previous medoid index, triangular inequality elimination criteria and partial distance search. Experimental results based on elliptic, curve and Gauss-Markov databases demonstrate that the proposed algorithm applied to $CLARANS$ may reduce the number of distance calculations by 67% to 92% while retaining the same average distance per object. In terms of the running time, the proposed algorithm may reduce computation time by 38% to 65% compared with the $CLARANS$ algorithm.

1 Introduction

The goal of clustering is to group sets of objects into classes such that similar objects are placed in the same cluster while dissimilar objects in separate clusters. Clustering (or classification) techniques are common forms of data mining [1] and have been applied in a number of areas including image compression [2], computer vision [3], psychiatry [4], medicine and marketing. A number of clustering algorithms have been proposed including k-means [5], k-medoids [6], $BIRCH$ [7], $CURE$ [8], $CACTUS$ [9], $CHAMELEON$ [10] and $DBSCAN$ [11]. Clearly, no single algorithm is suitable for all forms of input data, nor are all algorithms appropriate for all problems, however, the k-medoids algorithms

Y. Kambayashi, W. Winiwarter, M. Arikawa (Eds.): DaWaK 2002, LNCS 2454, pp. 63–72, 2002.

have been shown to be robust to outliers and are not generally influenced by the order of presentation of objects. Moreover, k-medoids algorithms are invariant to translations and orthogonal transformations of objects [6].

Partitioning Around Medoids (PAM) [6], *Clustering LARge Applications* ($CLARA$) [6] and *Clustering Large Applications based on RANdomized Search* ($CLARANS$) [12] are three popular k-medoids-based algorithms. In other work, our *Clustering Large Applications based on Simulated Annealing* ($CLASA$) algorithm applies simulated annealing to select better medoids [13]. Fuzzy theory can also be employed to develop fuzzy k-medoids algorithms [14] and while genetic algorithms can also be used [15]. The drawback of the k-medoids algorithms is the time complexity in calculating the medoids. However, there are many efficient algorithms developed for VQ-based clustering. These efficient codeword search algorithms used in VQ-based signal compression have not, to our knowledge, been applied to k-medoids-based algorithms.

In this paper, a novel and efficient k-medoids-based algorithm is proposed by using previous medoid index, triangular inequality elimination and partial distance search.

2 Existing Algorithms

2.1 *PAM* Algorithm

The k-medoids clustering algorithm evaluates a set of k objects considered to be representative objects (medoids) for k clusters within T objects such that the non-selected objects are clustered with the medoid to which it is the most similar. The total distance between non-selected objects and their medoid may be reduced by the swap of one of the medoids with one of the objects iteratively. The PAM (Partitioning Around Medoids) algorithm can be depicted as follows:

Step 1: Initialization - choose k medoids from T objects randomly.
Step 2: Evaluation - calculate the cost $D'_t - D_t$ for each swap of one medoid with another object, where D_t is the total distance before the swap and D'_t is the total distance after the swap.
Step 3: Selection - if the cost is negative, accept the swap with the best cost and go to step 2; otherwise record the medoids and terminate the program.

2.2 *CLARA* Algorithm

The computational complexity of the PAM algorithm is $O((1+\beta)k(T-k)^2)$ which is based on the number of partition per object, where β is the number of successful swaps. It can also be expressed as $O'((1+\beta)k^2(T-k)^2)$ based on the number of distance calculations. Obviously, PAM can be time consuming even for a moderate number of objects and small number of medoids. The $CLARA$ algorithm [6] reduces the computational complexity by drawing multiple samples of the objects and applying the PAM algorithm on each sample. The final