# Efficient Search Approaches for *K*-medoids-based Algorithms

Shu-Chuan Chu[1], John F. Roddick[1], Tsong-Yi Chen[2] and Jeng- Shyang Pan[2,3]

[1] School of Informatics and Engineering,
Flinders University of South Australia
[2] Department of Electronic Engineering,
National Kaohsiung University of Applied Sciences
[3] Department of Electrical and Electronic Engineering,
University of Adelaide

## Abstract

In this paper, the concept of previous medoid index is introduced. The utilization of memory for efficient medoid search is also presented. We propose a hybrid search approach for the problem of nearest neighbor search. The hybrid search approach is to combine the previous medoid index, the utilization of memory, the criterion of triangular inequality elimination and the partial distance search. The proposed hybrid search approach is applied to the k-medoids-based algorithms. Experimental results based on Gauss-Markov source, curve data set and elliptic clusters demonstrate that the proposed algorithm applied to *CLARANS* algorithm may reduce the number of distance calculation from 88.4% to 95.2% with the same average distance per object comparing with *CLARANS*. The proposed hybrid search approach can also be applied to the nearest neighbor searching and the other clustering algorithms.

## 1 Introduction

The clustering problems [1] have been investigated extensively in data mining, image compression, texture segmentation, computer vision, psychiatry, vector quantization, medicine and marketing. Recent works in the data mining community including *k*-medoids [2], *CACTUS* [3], *CHAMELEON* [4], and *AUTOCLUST+* [5]. No clustering algorithm described above has applied the concept of efficient codeword search in vector quantization area to reduce the computational complexity [6-9]. In this paper, we introduce the concept of previous medoid index and present the technique of the utilization of memory to reduce the computation time. We also present three search approaches for efficient clustering algorithms and apply these three search approaches to k-medoids-based algorithms that have been shown to be robust to outliers, not generally influenced by the order of presentation of objects and invariant to translations and orthogonal transformations of objects [2].

Partitioning Around Medoids (*PAM*) [2], Clustering LARge Applications (*CLARA*) [2] and Clustering Large Applications based on RANdomized Search (*CLARANS*) [10] are three popular *k*-medoids-based algorithms. Clustering Large Applications based on Simulated Annealing (*CLASA*) algorithm applied the simulated annealing to select better medoids [11]. The drawback of the k-medoids algorithms is the time complexity for getting the medoids. In this paper, three extended version of *CLARANS* are presented based on the proposed efficient search algorithms.

## 2 *K-medoids* 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 the $T$ 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 computational complexity of the *PAM* algorithm is $O((1+\beta)k^2(T-k)^2)$ which is based on the number of distance calculation and $\beta$ is the number of successful swap. Obviously, it is time consuming even for the middle number of objects and small number of medoids.

### 2.2 *CLARA* algorithm

*CLARA* (Clustering LARge Applications) algorithm [2] reduces the computational complexity by drawing multiple samples of the objects and applying *PAM* algorithm on each sample. The final medoids are obtained from the best result of these multiple draws. The computational complexity of the *CLARA* algorithm is $O(\alpha(k^2s^2 + k(T-k)) + \beta k^2 s^2)$, where $\alpha$, $s$, $k$, $\beta$ and $T$ are the number of samples, object size per sample, number of medoids, the number of successful swaps for all samples test and the total number of objects, respectively. Obviously, *CLARA* algorithm can deal with a large number objects than can be *PAM* algorithm if $s << T$. The *CLARA* algorithm can be depicted as follows:

**Step 1:** Repeat the following steps $q$ times.

**Step 2:** Call *PAM* algorithm with a random sample, $s$ objects from the original set of $T$ objects.

**Step 3:** Partition the $T$ objects based on the $k$ medoids obtained from previous step. Update the better medoids based on the average distance of the partition.

### 2.3 *CLARANS* algorithm

If the sample size $s$ is not large enough, the effectiveness (average distortion) of the *CLARA* algorithm is lower. However, the efficiency (computation time) is not good if the sample size is too large. It is tradeoff between the effectiveness and efficiency in *CLARA* algorithm. The best clustering cannot be obtained in *CLARA* algorithm if one of the best medoids is not included in the sample objects. In order to get the efficiency and acceptable performance in average distance per

0-7803-7490-8/02/$17.00©2002 IEEE.

object, the *CLARANS* (Clustering Large Applications based on RANdomized Search) algorithm [10] was proposed. The clustering process in *CLARANS* algorithm is formalized as searching through a certain graph where each node is represented by a set of $k$ medoids, and two nodes are neighbors if they only differ by one medoid. Each node has $k(T-k)$ neighbors, where $T$ is the total number of objects. *CLARANS* algorithm starts with a selected node randomly. It moves to the neighbor node if one test for the *maxneighbor* number of neighbors is successful; otherwise it records the current node as a local minimum. If the node with the local minimum is found, it starts with a new randomly selected node and repeat the search for a new local minimum. The procedure continues until the *numlocal* numbers of local minima have been found, and return the best node. The computational complexity is $O((\beta + numlocal)k(T-k))$ based on the number of distance calculation, where $\beta$ is the number of successful move between nodes. The *CLARANS* algorithm can be described as follows:

**Step 1:** Repeat the following steps for *numlocal* times.

**Step 2:** Select a current node randomly and calculate the average distance of this current code, where node is the collection of $k$ medoids.

**Step 3:** Repeat the following step for *maxneighbor* times.

— Select a neighbor node randomly and calculate the average distance for this node. If the average distance is lower, set current node to be the neighbor node.

# 3 Proposed search approaches

Although $k$-medoids-based algorithms are designed for clustering large database, all existing $k$-medoids-based algorithms are time consuming. The computational complexity of k-medoids-based algorithms can be reduced by applying the concept in VQ-based codeword search [6-9].

### 3.1 Partial distance search

The efficient codeword search algorithms in VQ-based signal compression have never been applied to $k$-medoids-based algorithms. The partial distance search (PDS) algorithm [6] is a simple and efficient codeword search algorithm which allows early termination of the distortion calculation between a test vector and a codeword by introducing a premature exit condition in the search process. Given the squared Euclidean distance measure, one object $x=\{x_1, x_2, ..., x_d\}$ and two medoids (representative objects) $o_t=\{o_{t1}, o_{t2}, ..., o_{td}\}$ and $o_j=\{o_{j1}, o_{j2}, ..., o_{jd}\}$, assume the current minimum distance is

$$D(x,o_t)=\sum_{i=1}^{d}(x_i-o_{ti})^2=D_{\min}, \quad (1)$$

$$\textit{if} \quad \sum_{i=1}^{h}(x_i-o_{ji})^2 \geq D_{\min}, \quad (2)$$

$$\textit{then} \quad D(x,o_j) \geq D(x,o_t), \quad (3)$$

where $1 \leq h \leq d$.

The efficiency of PDS is derived from the elimination of an unfinished distance computation if its partial accumulated distortion is larger than the current minimum distance. This will reduce $(d-h)$ multiplications and $2(d-h)$ additions at the expense of $h$ comparisons.

### 3.2 Triangular inequality elimination criteria

Vidal proposed the approximating and elimination search algorithm (AESA) [7] whose computation time is approximately constant for a codeword search in a large codebook size. The high correlation characteristics between data vectors of adjacent speech frames and the triangular inequality elimination (TIE) criterion were utilized to VQ-based recognition of isolated words [8].

Triangular inequality elimination (TIE) criterion is an efficient method for applying to nearest neighbor search. Let $o_1$ and $o_2$ be two different medoids and $x$ be an object, then TIE criterion cab be obtained as following.

$$\textit{If} \quad d(o_1,o_2) \geq 2d(x,o_1), \quad (4)$$

$$\textit{then} \quad d(x,o_2) \geq d(x,o_1). \quad (5)$$

In this criterion, these distances between all pairs of medoids are computed in advance. If Eq. 4 is satisfied, then we omit the computation of $d(x,o_2)$ if $d(x,o_1)$ has already been calculated. In this paper, TIE criterion is modified for a squared error distance measure. Given the medoid size $k$, a table with memory size $k(k-1)/2$ is made to store the one-fourth of squared distance between medoids,

$$\textit{if} \quad d^2(o_1,o_2)/4 \geq d^2(x,o_1), \quad (6)$$

$$\textit{then} \quad d(x,o_2) \geq d(x,o_1). \quad (7)$$

### 3.3 Previous medoid index

Most of the $k$-medoids-based algorithms are checked whether one of the medoids need to be changed by one of the objects. Since only one medoid is changed, most of the objects will belong to the cluster represented by the same medoid. By using this property, we may calculate the distance between the object and its previous medoid index firstly. Since the probability is very high for the object belong to the same medoid index, the distance is very small. If we get a very small distance between the object and one medoid, then it is easier to use *TIE* criterion and the partial distance search to reduce the distance computation.

### 3.4 Utilization of memory

Assume $k$ medoids $o_j, j=1,\cdots,k$, are chosen from $T$ objects $x_i, i=1,\cdots,T$ and the number of dimension for each object or medoid is $d$. The size of the memory for all objects in the database is $Td$. If the distance table for each pair of objects, $d(x_i,x_j)$, $i \neq j$, $i,j=1,\cdots,T$ are stored, then the size of memory for the distance table is $\frac{T(T-1)}{2}$. If this memory is available, then the distance calculation need be performed just the once, whether for *PAM*, *CLARA* and *CLARANS* algorithms. All these algorithms will thus be very efficient, and the computational complexity will be similar. Unfortunately, if the number of objects is large, memory is not always available. We thus propose a new approach which uses only $O(T-k)$ memory to store the distance, but it may reduce from $O((T-k)k)$ to $O(T-1+r(k-1))$, the distances computation for the test of the swap between object $o_{new}$ and $o_{old}$, where $r$ is the number of objects whose nearest medoids are swapped. The probability to swap the nearest medoid with any object is $1/k$, so $r \approx T/k$. Assume $NM(x_i)$ is the nearest medoid to the object $x_i$ before the swap, the total distance before the swap of object $o_{new}$ and medoid $o_{old}$ can be expressed as $D_t=\sum_{i=1}^{T-k} d(x_i, NM(x_i))$. The total distance after the swap of object $o_{new}$ and medoid

$o_{old}$ can be separated into three items. The first item is the distance for the objects which are not swapped from the medoids and their nearest medoids are not swapped to be the objects. This distance can be expressed as

$$D'_{i1} = \sum_{i=1}^{T-k} \min[d(x_i, NM(x_i)), d(x_i, o_{new})] \Big|_{\substack{x_i \neq o_{old} \\ NM(x_i) \neq o_{old}}}, \tag{8}$$

where $\min[d(x_i, NM(x_i)), d(x, o_{new})]$ denotes the minimum distance of $d(x_i, NM(x_i))$ and $d(x_i, o_{new})$. $x_i \neq o_{old}$ and $NM(x_i) \neq o_{old}$ represents that the object is not swapped from the medoid and the nearest medoid of the object is not swapped to be the object, respectively. The second distance is introduced from the medoids swapped to objects as following:

$$D'_{i2} = \min[d(o_{old}, o_j), \ j = 1, \cdots, k]. \tag{9}$$

The third distance is introduced from those objects whose nearest medoids are swapped to be objects as following:

$$D'_{i3} = \sum_{i=1}^{T-k} d(x_i, o_p) \Big|_{\substack{x_i \neq o_{old} \\ NM(x_i) = o_{old} \\ x_i \in S_p}}, \tag{10}$$

where $NM(x_i) = o_{old}$ represents those objects whose nearest medoids are swapped to be objects and $S_p$ is the $p$th partitioned set where $o_p$ is the representative medoid. Hence the total distance after the swap of object $o_{new}$ and medoid $o_{old}$ can be expressed as

$$D'_i = \sum_{i=1}^{T-k} \{\min[d(x_i, NM(x_i)), d(x_i, o_{new})] \Big|_{\substack{x_i \neq o_{old} \\ NM(x_i) \neq o_{old}}} + d(x_i, o_p) \Big|_{\substack{x_i \neq o_{old} \\ NM(x_i) = o_{old} \\ x_i \in S_p}} \} + \min[d(o_{old}, o_j), \ j = 1, \cdots, k]. \tag{11}$$

If the distances $d(x_i, NM(x_i)), \ i = 1, \cdots, T-k$ are stored, then only $(T-k-1-r)$ distances computation for $d(x_i, o_{new}), i = 1, \cdots, T-k$, $x_i \neq o_{old}$, $NM(x_i) \neq o_{old}$, and $k$ distances computation for $d(o_{old}, o_j)$, $j = 1, \cdots, k$ and $rk$ distances computation for $d(x_i, o_p), i = 1, \cdots, T-k$, $NM(x_i) = o_{old}$. Since the memory size $(T-k)$ is generally reasonable for the clustering of the objects with memory size $Td$, it is a useful approach. Note that this approach can be applied to *PAM*, *CLARA*, *CLARANS* and the other clustering algorithms.

### 3.4 Proposed search approaches

In this paper, three new search approaches are presented for the problem of nearest neighbor search. These three new search approaches are applied to *CLARANS* algorithm. *CLARANS* algorithm with previous medoid index, the criterion of TIE and PDS is referred to as *CLARANS—ITP*. *CLARANS* with the proposed utilization of memory is referred to as *CLARANS—M*. Application of the previous medoid index, the proposed utilization of memory, the first criterion of TIE and partial distance search algorithm to CLARANS is referred to as *CLARANS—MITP*.

## 4. Experimental Results

### 4.1 data sets

Three artificial data sets were used for the experiments as follows:

1. 3,000 objects with 8 dimensions are generated from the Gauss-Markov source which is of the form $y_n = \alpha y_{n-1} + w_n$ where $w_n$ is a zero-mean, unit variance, Gaussian white noise process, with $\alpha = 0.5$.
2. 12,000 objects with 2 dimensions collected from twelve elliptic clusters.
3. 5,000 objects with 2 dimensions are generated from curve data sets. The object $(x, y)$ is collected from the form $-2 \le x \le 2$ and $y = 8x^3 - x$.

### 4.2 Experiments

In this paper, three extended version of *CLARANS* are presented. Experiments were carried out to test the number of distances calculation and the average distance per object for *CLARA*, *CLARANS*, *CLARANS—ITP*, *CLARANS—M* and *CLARANS—MITP* algorithms. Since the computation time depends not only on the clustering algorithm but also on the use of computation facility. It is better to choose one measure criterion so that the measure results are the same for all types of computers and this measure criterion is proportional to the computation time. That is why we choose the number of distance calculation as the benchmark. Squared Euclidean distance measure is used for the experiments. The Gauss-Markov source was used for the first experiment. 32 medoids are selected from 3000 objects. For *CLARA* algorithm, the parameter $q$ was set to 5 and $s$ was set to $32+2*k$ for the sample size, where $k$ is the number of medoids. For *CLARANS* algorithm, the parameters *numlocal* and *maxneighbor* are set to 5 and 1200, respectively. Experimental results are shown in Table 1, comparing with *CLARANS*, *CLARANS—MITP*, *CLARANS—M* and *CLARANS—ITP* may reduce the number of distance computation by 95.2%, 93.8% and 67%, respectively.

Table 1. Results of Experiment for Gauss-Markov source

| Seed | CLARA | | CLARANS | CLARANS-ITP | CLARANS-M | CLARANS-MITP | |
|---|---|---|---|---|---|---|---|
| | Count of dis. ($10^7$) | Ave. dis. | Count of dis. ($10^7$) | Count of dis. ($10^7$) | Count of dis. ($10^7$) | Count of dis. ($10^7$) | Ave. dis. |
| 1 | 1548 | 4.559 | 376 | 126 | 23 | 18 | 4.432 |
| 2 | 1637 | 4.592 | 532 | 177 | 33 | 25 | 4.359 |
| 3 | 1789 | 4.551 | 375 | 123 | 23 | 18 | 4.381 |
| 4 | 1726 | 4.578 | 406 | 133 | 25 | 19 | 4.398 |
| 5 | 1827 | 4.559 | 397 | 131 | 25 | 19 | 4.367 |
| 6 | 1675 | 4.526 | 414 | 137 | 26 | 20 | 4.384 |
| 7 | 1853 | 4.527 | 323 | 106 | 20 | 15 | 4.380 |
| 8 | 1624 | 4.483 | 396 | 130 | 25 | 19 | 4.393 |
| 9 | 1903 | 4.545 | 367 | 120 | 23 | 17 | 4.377 |
| 10 | 1802 | 4.514 | 358 | 119 | 22 | 17 | 4.406 |
| Ave. | 1738 | 4.543 | 394 | 130 | 24 | 19 | 4.388 |

The twelve elliptic clusters were used for the second experiment. 12 medoids are selected from 12000 objects. For *CLARA* algorithm, the parameter $q$ was set to 5 and $s$ was set to $960+2*k$. For *CLARANS* algorithm, the parameters *numlocal* and *maxneighbor* are set to 5 and 1800, respectively. As shown in Fig. 1, comparing with *CLARANS*, *CLARANS—MITP*, *CLARANS—M* and *CLARANS—ITP* may reduce the number of distance computation by 88.4%, 84%

and 84.3%, respectively.

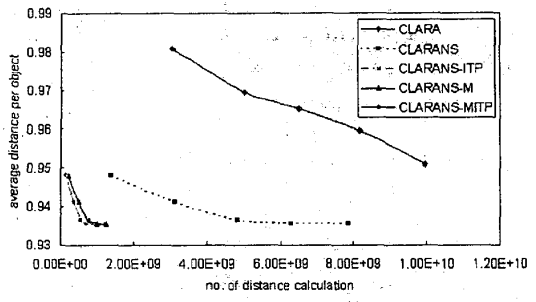


Fig. 1 Performance comparison of *CLARA*, *CLARANS*, *CLARANS-MITP*, *CLARANS-M* and *CLARANS-ITP* for twelve elliptic clusters

The curve database was used for the third experiment. 20 medoids are selected from 5000 objects. For *CLARA* algorithm, the parameter $q$ was set to 5 and $s$ was set to $400+2*k$. For *CLARANS* algorithm, the parameters *numlocal* and *maxneighbor* are set to 5 and 1250, respectively. As shown in Fig. 2 and Table 2, comparing with *CLARANS*, *CLARANS-MITP*, *CLARANS-M* and *CLARANS-ITP* may reduce the number of distance computation by 93.9%, 90.2% and 92%, respectively.

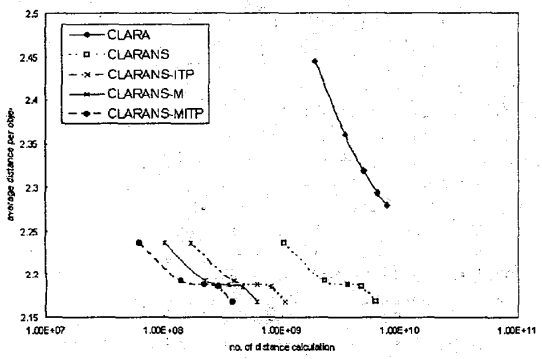


Fig. 2 Performance comparison of *CLARA*, *CLARANS*, *CLARANS-MITP*, *CLARANS-M* and *CLARANS-ITP* for curve clusters

Table 2. Results of Experiment for curve clusters

| Seed | CLARA | | CLARANS | CLARANS-ITP | CLARANS-M | CLARANS-MITP | |
|---|---|---|---|---|---|---|---|
| | Count of dis. ($10^7$) | Ave. dis. | Count of dis. ($10^7$) | Count of dis. ($10^7$) | Count of dis. ($10^7$) | Count of dis. ($10^7$) | Ave. dis. |
| 1 | 741 | 2.232 | 652 | 53 | 64 | 40 | 2.151 |
| 2 | 734 | 2.278 | 626 | 51 | 61 | 38 | 2.181 |
| 3 | 847 | 2.317 | 702 | 57 | 69 | 43 | 2.179 |
| 4 | 797 | 2.342 | 719 | 58 | 70 | 44 | 2.192 |
| 5 | 840 | 2.321 | 574 | 45 | 56 | 35 | 2.156 |
| 6 | 805 | 2.252 | 623 | 49 | 61 | 38 | 2.157 |
| 7 | 741 | 2.305 | 650 | 52 | 64 | 39 | 2.171 |
| 8 | 797 | 2.238 | 636 | 51 | 62 | 39 | 2.144 |
| 9 | 741 | 2.209 | 551 | 44 | 54 | 34 | 2.185 |
| 10 | 805 | 2.317 | 484 | 38 | 47 | 29 | 2.165 |
| Ave. | 785 | 2.281 | 622 | 50 | 61 | 38 | 2.168 |

## 5 Conclusions and Future Work

In this paper, three extended version of *CLARANS* are presented based on the proposed three search strategies. Experimental results demonstrate that applying the proposed hybrid search method using previous medoid index, utilization of memory, the criterion of TIE and partial distance search to *CLARANS* may reduce the number of distance computation from 88.4% to 95.2% comparing with *CLARANS*. Note that the proposed search strategies may apply to the other clustering algorithms. We will extend the work of this paper to some other well-known clustering algorithms for future work.

## References


[1] J. F. Roddick and M. Spiliopoulou, "A Survey of Temporal Knowledge Discovery Paradigms and Methods," in *IEEE Transactions on Knowledge and Data Engineering*, 2002.

[2] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: An Introduction to Cluster Analysis*, New York: John Wiley & Sons, 1990.

[3] V. Ganti, J. Gehrke and R. Ramakrishnan, "CACTUS-Clustering categorical data using summaries," in *International Conference on Knowledge Discovery and Data Mining*, (San Diego, USA), pp. 73-83, 1999.

[4] G. Karypis, E. –H. Han and V. Kumar, "Chameleon: A hierarchical clustering algorithm using dynamic modeling," *Computer*, pp. 32-68, 1999.

[5] V. Estivill-Castro and I. Lee, "AUTOCLUST+: automatic clustering of point-data sets in the presence of obstacles", *First International Workshop on Temporal, Spatial and Spatial-Temporal Data Mining*, pp. 133-146, 2000.

[6] C. D. Bei and R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization," *IEEE Trans. Commun.*, vol. COM-33, no. 10, pp. 1132-1133, 1985.

[7] E. Vidal, "An algorithm for finding nearest neighbours in (approximately) constant average time, " *Pattern Recognition Letters*, vol. 4, pp. 145-157, 1986.

[8] S. H. Chen and J. S. Pan, "Fast search algorithm for VQ-based recognition of isolated word, " *IEE Proc. I*, vol. 136, no. 6, pp. 391-396, 1989.

[9] J. S. Pan, F. R. McInnes and M. A. Jack, "Bound for Minkowski metric or quadratic metric applied to VQ codeword search, " *IEE Proc. Vision Image and Signal Processing*, vol. 143, no. 1, pp. 67-71, 1996.

[10] R. Ng and J. Han, "Efficient and effective clustering method for spatial data mining," in *Twentieth International Conference on Very Large Data Bases*, (J. B. Bocca, M. Jarke, and C. Zaniolo, eds.), (Santiago, Chile), pp. 144-155, Morgan Kaufmann, 1994.

[11] S. C. Chu, J. F. Roddick and J. S. Pan, "A comparative study and extensions to k-medoids algorithms," in *Fifth International Conference on Optimization: Techniques and Applications*, (Hong Kong, China), pp. 1708-1717, 2001.