Robust Clustering based on Winner-Population Markov Chain

Fu-Wen Yang[†], Hwei-Jen Lin[*], Patrick S. P. Wang, and Hung-Hsuan Wu[*]
[†][*]*Department of Computer Science and Information Engineering,*
Tamkang University, Taipei, Taiwan, ROC
College of Computer and Information Science, Northeastern U., Boston, MA, USA
[†]*112405@mail.tku.edu.tw*

Abstract

In this paper, we propose an unsupervised genetic clustering algorithm, which produces a new chromosome without any conventional genetic operators, and instead according to the gene reproducing probabilities determined by Markov chain modeling. Selection of cluster centers from the dataset enables construction of a look-up table that saves the distances between all pairs of data points. The experimental results show that the proposed algorithm not only solves the premature problem to provide a more stable clustering performance in terms of number of clusters and clustering results, but also improves the time efficiency.

1. Introduction

Clustering is a useful technique for the applications in image segmentation, information retrieval, pattern recognition, data mining, and machine learning. However, in many such problems, there is little prior information and few assumptions about the data (cluster shapes, number of clusters, initial conditions, etc.). Several algorithms require information for clustering, such as K-means, Fuzzy-c-means, EM, etc, as previous literature has stated [1-2]. However, the number of clusters of a data set is not given as prior information in most real life situations and these clustering systems are not able to automatically and efficiently form nature groups of the input patterns in these situations. The clustering problem in such situations is referred to unsupervised clustering. In the research of unsupervised clustering, the evolutionary approaches are often employed and provide good clustering results. Such approaches can automatically determine optimal number of clusters. Genetic algorithms (GAs) are the best-known evolutionary techniques [3-4]. To date, some research articles have dealt with these methods [5-9]. Among the GA-based clustering algorithms illustrated in the current literature, the **GCUK** (Genetic Clustering for Unknown K) method proposed by Bandyopadhyay and Maulik [9] is one of the most effective. However, it is very time-consuming due to its usage of string representation (or real-number encoding) for encoding clusters, which require a great deal of time for floating-point computation. In our previous paper [10], we proposed an unsupervised clustering method, called the **PMCC** algorithm, that outperforms the **GCUK** method in terms of both time efficiency and the clustering results. The **PMCC** algorithm, based on population Markov chain [10], uses the gene reproducing probabilities of Markov chain modeling to perform evolution without any genetic operations, so that it saves a great deal of computational time required by the canonical genetic operations. Selection of cluster centers from the dataset enables construction of a look-up table that saves the distances between all pairs of data points, and thus the repeated evaluation of fitness during the evolution process can be avoided. Nevertheless, even though the **PMCC** algorithm behaves quite well when compared with the **GCUK** method, it still has the problem of premature convergence, especially when the number of clusters included in the data set tends to be large. This was our motivation to propose an improved version of the **PMCC** method: the **WPMCC** (Winner Population Markov Chain) method. The results of our experiments show that this improved version not only solves the premature convergence problem providing a more stable clustering performance in terms of number of clusters and clustering results, but it also improves the time efficiency.

This paper is organized as follows: Section 2 illustrates the preliminary of the canonical genetic algorithms. In Section 3, the proposed clustering algorithm based on winner-population Markov chain is introduced. Experimental results and discussion are given in Section 4, with our conclusion in Section 5.

2. Preliminary

Genetic algorithms are search and optimization algorithms based on the principles of natural evolution. They have been frequently used in unsupervised clustering. In many theoretical studies of GAs [11-13], the population Markov chain models have been adopted. Yong Gao et al. [13] proposed a novel genetic algorithm (called **GANGO2**) which needs neither to maintain a population nor to use the conventional genetic operators, and yet has the same search mechanisms as the classical GAs. They can be implemented by directly sampling the transition probability distributions instead of applying the conventional genetic operators to evolve the populations. The theoretical analyses and their proposed theorem are introduced in this Section.

Definition: Given a population X = (X'$_1$, …, X'$_P$), X'$_i$ = (x$_{i1}$, …, x$_{il}$), i = 1,…, P, for any positive integer 1≤ $j \leq l$, let I^j_0 and I^j_1 denote the sets of indices of all the chromosomes of the population X that have respectively a zero or one at the j-th gene position, that is, $I^j_0 = \{x_{ij} = 0, 1 \leq i \leq P\}$, $I^j_1 = \{x_{ij} = 1, 1 \leq i \leq P\}$,

$$F(X) = \sum_{i=1}^{N} f(X'_i),\ F^j_0(X) = \sum_{i \in I^j_0} f(X'_i),\ F^j_1(X) = \sum_{i \in I^j_1} f(X'_i),$$

$$a_j = \frac{F^j_0(X)}{F(X)} \quad , \qquad b_j = \frac{F^j_1(X)}{F(X)} = 1 - a_j \tag{1}$$

Theorem: Consider the GA population Markov chain {X(k), generation k≥0}. Given X(k) = X, the conditional distribution of the j-th component x$_{ij}$(k+1) of individual X'$_i$(k+1) is a zero-one distribution with the parameter uniquely determined by the characteristic of X and the mutation probability p$_m$ as

$$p_j(k+1,0) = P\{x_{ij}(k+1) = 0 | X(k) = X\} = a_j + (1-2a_j)p_m \quad (2)$$

$$p_j(k+1,1) = P\{x_{ij}(k+1) = 1 | X(k) = X\} = b_j + (1-2b_j)p_m \quad (3)$$

Although the over-all performance of our previously proposed clustering algorithm, called **PMCC,** based on **GANGO2** is fine, it still has some problems: (1) Although the fitter chromosome can immediately contribute to the creation of the other chromosomes of the later population, the initial population sometimes tends to influence the outcome during the entire evolution process. (2) The values of $F(X(k+1))$ and $F^j_1(X(k+1))$ tend to unrestrictedly expand, and the effects will decay in the later and fitter chromosomes. (3) The average threshold, $t(k+1)$, is a cumulative sum of the fitness values from duplicate individuals, so the use of this threshold tends to prematurely converge, especially when the dataset has more than 7 clusters.

3. The Proposed Clustering algorithm

This section describes in more depth how the proposed method is implemented.

3.1. Binary Representation

The cluster centers are selected from the data set. The chromosome length is equal to the size of the data set. The j-th gene of a chromosome corresponds to the j-th data point in the data set. If the j-th data point is selected to be a cluster center, the allele of the j-th gene in the chromosome is set to "1"; otherwise "0". The number of clusters, denoted by K, is assumed to lie in the range [K$_{min}$, K$_{max}$], where K$_{min}$ is set to 2, and K$_{max}$ is commonly set to $N/2$ or $\sqrt{N}$, where N is the chromosome length (or the size of the input data), unless otherwise specified.

3.2. Population Initialization

Let P be the population size. First, an integer K$_r$ for the r-th chromosome, r = 1, 2, …, P, is randomly selected from the range [K_{min}, K_{max}], and then K$_r$ distinct data points are randomly chosen from the data set, the allele of the gene corresponding to the index of each of the chosen data points is set to "1"; while that of each of the remaining genes is set to "0". For example, if $N = 16$, $K_r = 3$ for the r-th chromosome, and 3 data points randomly chosen from the data set have indices 3, 10, and 12, respectively, then the chromosome should be 0010 0000 0101 0000.

3.3. Fitness Function Evaluation

The clustering results should have the following properties: (1) homogeneity within the clusters and (2) heterogeneity between clusters. To evaluate the clustering results, several cluster validity measures have been proposed [1, 14, 15]. We employed the Davies-Bouldin index (DB index) [14] to measure the validity of the clusters, since our experiments showed that the DB index is better than other indices such as the Dunn index and the XB index. As given in Equation (6), the DB index is a function of the ratio of the sum of the within-cluster scatter to the between-cluster separation, which provides an appropriate measurement. In Equations (4) and (5), $S_{i,q}$ denotes the measure of dispersion of a cluster C_i, $i = 1, \ldots, k$, appearing in a chromosome Ch. $R_{i,qt}$ denotes the maximal similarity index of C_i to the other clusters and $d_{ij,t} \equiv d(C_i, C_j)$ denotes the Minkowski distance of order t between C_i and C_j (q = 1 and t = 2 in this paper.) As given in Equation (7), the fitness function for our proposed algorithm is defined as the reciprocal of the DB index.

$$S_{i,q} = \left(\frac{1}{|C_i|} \sum_{x \in C_i} \| x - z_i \|_2^q \right)^{1/q}, \tag{4}$$

where z_i is the center of the cluster C_i

$$R_{i,qt} = \underset{j,j \neq i}{Max} \{ (S_{i,q} + S_{j,q}) / d_{ij,t} \}, \tag{5}$$

where $d_{ij,t} = d(C_i, C_j) = \| z_i - z_j \|_t$

$$DB = \frac{1}{k} \sum_{i=1}^{k} R_{i,qt} \tag{6}$$

$$Fitness\ (Ch) = \frac{1}{DB} \tag{7}$$

3.4. Winner-Population Markov Chain Clustering Algorithm

The winner-population Markov chain clustering algorithm (**WPMCC)** is given as follows:

```
Step 1. Set $k \leftarrow 0$, and generate initial population $X(0) = \{X'(1), X'(2), \ldots, X'(P)\}$, compute $F(X(0))$,
        $F^j_1(X(0))$, $b_j(k)$, and $p_j(k, 1)$, $1 \le j \le l$, according
        to Eq.s (1 & 3), and set $t(0) \leftarrow \underset{1 \le i \le P}{Max} \{f(X'(i))\}$.
Step 2. //Initializing $F(X(k+1))$ and $F^j_1(X(k+1))$
        $F(X(k+1)) \leftarrow t(k)$,  $t(k+1) \leftarrow t(k)$,
        for $j \leftarrow 1$ to $l$ do
          $F^j_1(X(k+1)) \leftarrow b_j(k) \times F(X(k+1))$
Step 3. //Generating a new population
        for $i \leftarrow 1$ to $C$ do
          Independently sample $p_j(k, 1)$, $1 \le j \le l$, to get
            a chromosome $X'(i) \leftarrow (x_1(i), x_2(i), \ldots, x_l(i))$.
          if ( $f(X'(i)) > t(k)$ ) then
            if ($t(k+1) < f(X'(i))$ then $t(k+1) \leftarrow f(X'(i))$,
            //update $F(X(k+1))$ and $F^j_1(X(k+1))$
            $F(X(k+1)) \leftarrow F(X(k+1)) + f(X'(i))$
            for $j \leftarrow 1$ to $l$ do
              if $x_j(i) = 1$ then
                $F^j_1(X(k+1)) \leftarrow F^j_1(X(k+1)) + f(X'(i))$
Step 4. If some stopping criterion is met then stop
        else for $j \leftarrow 1$ to $l$ do
                compute $b_j(k+1)$ and $p_j(k+1, 1)$,
             $k \leftarrow k + 1$ and go to Step 2.
```

For providing more stable clustering results, we count the accumulative sum of the probabilities of population Markov chain modeling for each gene in a population of *C* chromosomes. If we set *C* equal to *1*, the **WPMCC** algorithm becomes similar to the **PMCC** algorithm. That is, the fitter chromosomes may immediately contribute to the creation of the other chromosomes in the later population. This causes quick convergence and yields unstable results. Conversely, the greater the value of *C* is, the more slowly the **WPMCC** algorithm converges and more stable results it provides. For preventing the premature convergence, first, we use the maximum fitness value as the threshold for each population of *C* chromosomes. Only the chromosomes with fitness greater than the threshold can affect and change the values of $F(X(k+1))$ and $F^j_1(X(k+1))$. In such a way, these values would not be unlimitedly affected by the same individuals again and again. Second, we initialize the values of $F(X(k+1))$ and $F^j_1(X(k+1))$ for each generation to avoid unlimited expansion when they are modified in Step 2. Because chromosomes greater than the threshold become fewer and fewer, any chromosome produced in the later generations contributes more and more effect.

4. Experimental Results

The experiments were implemented in an environment using the Intel Centrino-Mobile 1.3GHz CPU, 30G HDD, 256M RAM and Microsoft Windows XP. In our experiments, *100* artificial and random data sets with a variety of numbers (in $[K_{min}, K_{max}] = [2, 11]$) of clusters were tested to evaluate the performance of the proposed method. These data sets are publicly available on the Website: http://pria.cs.tku.edu.tw. In our experiments, p_m is automatically estimated by the equation $p_m \approx 1.75/(P \times \sqrt{l})$, $p_c = 0.9$ as required in [16], $P = C = 100$, $G = 100$, and $[K_{min}, K_{max}] = [2, \sqrt{N}]$. Finally, the DB index was adopted to measure the validity of the clusters. For comparison, we performed both our methods and the **GCUK** method *10* runs on each data set. Figure 1 shows the average maximum fitness values resulting from these methods, having been tested *10* runs for each data set, respectively. It demonstrates that on the average the **WPMCC** algorithm indeed provides better fitness values than any of the other methods, especially when the dataset has more than *5* clusters. Figure 2 shows the average processing time per data point required by each method tested *10* runs for each data set, and demonstrates that the **WPMCC** algorithm is about *3* to *7* times faster than the **GCUK**-clustering method and a little bit faster than the **PMCC** method. Our experiments also show that the **WPMCC** algorithm converges before the *15*th generation and has greater maximum fitness values than any of the others.

5. Conclusions

This paper modifies the previously proposed unsupervised clustering **PMCC** algorithm, to achieve an improved version: the **WPMCC** algorithm, which not only improves the premature convergence problem so as to provide a more stable clustering performance, but also improves the time efficiency. Using the Euclidean distance as the dissimilarity metric yields circular clusters. Such clusters for some of the test data may not as natural as those provided by people. In the future, we will test the other distance metric such as Mahalanoobis distance and point symmetry distance [17] against a variety of data sets with various shapes of clusters. In addition we are investigating the correlation between the convergence speed and the number of clusters in the data set and studying on similarity/dissimilarity metrics and expect to further improve the unsupervised clustering algorithm.

6. References

[1] S. Theodoridis and K. Koutroumbas, Pattern Recognition, Academic Press, New York, 1999.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn, Data Clustering: A Review, ACM Computing Surveys, Vol. 31, No. 3, 1999, 264-323.

[3] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, New York, 1989.

[4] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, New York, 1992.

[5] V. V. Raghavan and K. Birchand, A clustering strategy based on a formalism of the reproductive process in a natural system, Proceedings of the 2nd International Conference on Information Storage and Retrieval, 1979, 10-22.

[6] D. Jones and M. A. Beltramo, Solving partitioning problems with genetic algorithms, Proceedings of the 4th International Conference on Genetic Algorithms, 1991, 442-449.

[7] G. P. Babu and M. N. Murty, Clustering with evolution strategies, Pattern Recognition, 27, 1994, 321-329.

[8] Ujjwal Maulik and Sanghamitra Bandyopadhyay, Genetic algorithm-based clustering technique, Pattern Recognition, 33, 2000, 1455-1465.

[9] Sanghamitra Bandyopadhyay and Ujjwal Maulik, Genetic clustering for automatic evolution of clusters and application to image classification, Pattern Recognition, 35, 2002, 1197-1208.

[10] Hwei-Jen Lin, Fu-Wen Yang and Yang-Ta Kao, Efficient Clustering based on Population Markov Chain, Proceedings of the IASTED International Conference on Modeling, Simulation and Optimization (ICMSO2004), 2004, 117-123.

[11] Y. Gao, Z. B. Xu, and G. Li, Theoretical analyses, new algorithms, and some applications of genetic algorithms: A review of some recent work, in Fuzzy Logic and Soft Computing, K. Y. Cai, G. Chen and M. Ying, Kluwer Academic, New York, 1999, 121-134.

[12] Y. Leung, Y. Gao, and Z. B. Xu, Degree of population diversity: A perspective on premature convergence in genetic algorithms and its Markov chain analysis, IEEE Trans. Neural Networks, Vol. 8, No. 5, 1997, 1165-1176.

[13] A. E. Eiben, E. H. L. Arts, and K. M. Van Hee. Global convergence of genetic algorithms: A Markov chain analysis, in Parallel Problem Solving from Nature, H. P. Schwefel and R. Manner, Springer, Berlin and Heideberg, 1991, 4-12.

[14] D. L. Davis and D. W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 1, No. 4, 1979, 224-227.

[15] J. C. Bezdek, Some new indexes of cluster validity, IEEE Trans. Systems, Man, and Cybernetics—Part B: Cybernetics, Vol. 28, No. 3, 1998, 301-315.

[16] J. D. Schaffer, R. A. Caruna, L. J. Eshelman, and R. Das, A Study of control parameters affecting online performance of genetic algorithms for function optimization, in Proceedings of the 3rd International Conference on Genetic Algorithms and Their Applications, San Mateo, CA: Morgan Kaufman, 1989, 51-60.

[17] Mu-Chun Su and Chien-Hsing Chou, A Modified Version of the K-Means Algorithm with a Distance Based on Cluster Symmetry, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 23, No. 6, June 2001, 674-680.

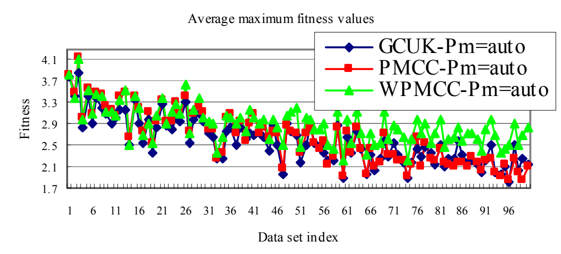

Figure 1. Average maximum fitness value

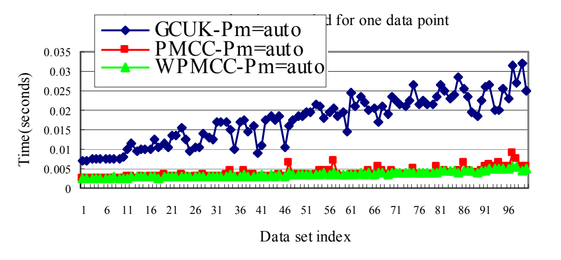

Figure 2. Average processing time required by each data point