2003 International Symposium on Advanced Intelligent Systems
September 25-28, 2003, Jeju, Korea

A Hybrid Genetic Algorithm for K-Means Clustering

*Sung-Hae Jun, **Jin-Woo Han, **Minjae Park, **Kyung-Whan Oh

*Dept. of Statistics, Chongju University

36, Naedok-Dong, Sangdang-Gu, Cheongju, Chungbuk, Korea (360-764)

email:shjun@chongju.ac.kr

**Dept. of Computer Science, Sogang University

1 Shinsoo-Dong, Mapo-Gu, Seoul, Korea (121-742)

email:{hey_han@ailab, pmj219@ailab, kwoh@ccs}.sogang.ac.kr

Abstract **- Initial cluster size for clustering of partitioning methods is very important to the clustering result. In K-means algorithm, the result of cluster analysis becomes different with optimal cluster size K. Usually, the initial cluster size is determined by prior and subjective information. Sometimes this may not be optimal. Now, more objective method is needed to solve this problem. In our research, we propose a hybrid genetic algorithm, a tree induction based evolution algorithm, for determination of optimal cluster size. Initial population of this algorithm is determined by the number of terminal nodes of tree induction. From the initial population based on decision tree, our optimal cluster size is generated. The fitness function of ours is defined an inverse of dissimilarity measure. And the bagging approach is used for saving computational time cost**

I. INTRODUCTION

Many clustering algorithms have been introduced. Among them, K-means algorithm especially requires an initial number of clusters to be determined. But this determination of cluster size is subjective and the accuracy of the clustering results may depend on the initial cluster size. In this paper, we'll determine the optimal cluster size using hybrid genetic algorithm. This number is used for k for k-means algorithm. Moreover, this determination of cluster size is needed in clustering like hierarchical clustering method, etc. In this paper, we'll propose an objective method to determine the optimal cluster size using our proposed method. In next section, we'll introduce k-means clustering and initial cluster size. In section III, proposed hybrid genetic algorithm for k-means algorithm. Our experimental result for the proposed methods will be shown in section IV. Final section will give conclusion of our research and the direction of future study. This paper will contribute to the selection of optimal cluster size. In our experiment, we showed results of optimal determination of cluster size using training data of UCI Machine Learning Repository.

II. K-MEANS ALGORITHM AND INITIAL CLUSTER SIZE

Input vectors of n-dimensions may be considered as representing points within an n-dimensional Euclidean space. The K-means algorithm is one of many clustering techniques that partakes of this notion of clustering by minimum distance. The Euclidean metric of K-means algorithm is defined as follows.

$$\| x \| = \left[\sum_{i=1}^{n} x_i^2 \right]^{1/2} \qquad (1)$$

where, x is $(x_1, x_2, ..., x_n)^T$. (1) is the norm of the input vector x. The K-means algorithm is implemented in the following steps[Pandya, 1995].

(Step 1) Initialize

Choose the number of cluster, k.

$$\{c_1(l), c_2(l), ..., c_k(l)\}$$

$c_i(l)$: the value of the cluster center at the l th iteration. The starting value can be arbitrary.

(Step 2) Attach objects

Each object vector $x^{(p)}$ is attached to one of the K clusters according to the following criteria.

$$x^{(p)} \in S_j(l) \text{ if } \| x^{(p)} - c_j(l) \| < \| x^{(p)} - c_i(l) \|$$

for all i=1,2,...,K, i≠j

$S_j(l)$: the population of cluster j at iteration l.

(Step 3) Calculate new cluster centers

Using the new cluster sets of step 2, recalculate the value of each cluster center such that the sum of the distances from each member vector to the new cluster center is minimized. So we wish to minimize J_j.

$$J_j = \sum_{x^{(p)} \in S_j(l)} \| x^{(p)} - c_j(l+1) \|^2$$

j=1,2,...,K

$$c_j(l+1) = \frac{1}{N_j} \sum_{x^{(p)} \in S_j(l)} x^{(p)}$$

where, N_j is the number of object vectors attached to S_j during step 2.

(Step 4) Check for convergence

The condition of convergence is that no cluster center has changed its position during step 3. This can be represented to the following.

$$c_j(l+1) = c_j(l)$$

j=1,2,...,K

If this equation is satisfied, then convergence has occurred. Otherwise iterate by going to step 2.

In the above, the K of (Step 1) in K-means algorithm is chosen arbitrary. But this K is important to clustering results. So, we have researched on the determination of optimal cluster size[5][6].

III. HYBRID GENETIC ALGORITHM FOR K-MEANS CLUSTERING

A. Genetic algorithm

Genetic algorithm(GA) was invented by John Holland in the 1960s. Holland's GA is a method for moving from one population of 'chromosomes' (e.g., strings of ones and zeros, or 'bits') to a new population by using a kind of 'natural selection' together with the genetics-inspired operators of crossover, mutation, and inversion. GA was well suited for some of the most pressing computational problems re-quire searching through a huge number of possibilities for solutions. Biological evolution is an appealing source of inspiration for addressing theses problems. Evolution is, in fact, a method of searching among an enormous number of possibilities for solutions. The rules of evolution are remarkably simple: species evolve by means of random variation (via mutation, recombination, and other operators), followed by natural selection in which the fittest tend to survive and reproduce, thus propagating their genetic material to future generations[1].

B. Hybrid genetic algorithm for optimal clustering

In our research, the initial population of genetic algorithm is determined by decision tree. Decision tree is an attribute chart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions[3][9]. The topmost node in a tree is the root node. In detail, our used decision tree is a regression tree. This can be used for classification and prediction. A regression tree is similar to a decision tree in the sense that tests are performed at the internal nodes. A major difference is at the leaf level-while in a decision tree a majority voting is per-formed to assign a class label to the leaf, in a regression tree the mean of the objective attribute is computed and used as the predictive value[10]. In this paper, the initial population of GA for optimal clustering is determined by the number of terminal nodes in trained decision tree. Decision tree is an attribute chart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions[3][12]. The topmost node in a tree is the root node. In detail, our used decision tree is a regression tree. This can be used for classification and prediction. A regression tree is similar to a decision tree in the sense that tests are performed at the internal nodes. A major difference is at the leaf level-while in a decision tree a majority voting is performed to assign a class label to the leaf, in a regression tree the mean of the objective attribute is computed and used as the predictive value[11]. Our regression tree is designed as Table 1.

Table 1. The option of used regression tree in our paper

Options	Our determination
Splitting Criterion	F-test (P-value: 0.2)
Maximum number of branches from a node	1
Maximum number of branches from a node	2
Maximum depth of tree	6

In Table 1, target variable of regression tree is decided as the input variable with minimum variance[4]. The proposed GA for determination of clustering number comprise following main six components.

1) **Description of an Individual**: Each individual represent a possible solution to the problem and is composed of string of genes. In our proposed GA, each individual was represented the number of cluster and coded in 6-bit binary string.

2) **Initial Population**: The approximated cluster number was taken by DT based approach. This number was set as the center of initial population. The around numbers of center were assigned as other members of initial population. The size of initial population is eleven

3) **Fitness Function**: Our proposed fitness function is defined follows:

$$F = \begin{cases} 0 & if\ K = 0 \\ \dfrac{1}{e^{\frac{c}{k}}} \sum_{i=1}^{n} Correlation(e_i, C_i) & otherwise \end{cases}$$

Where, e_i = ith element in data set, C_i = centroid of cluster that holds ith element,

K = Number of clusters, c = Constant

There are two parts in this function.

$\sum_{i=1}^{n} Correlation(e_i, C_i)$: The correlation of all clusters

$\frac{1}{e^{\frac{c}{k}}}$: The penalty of cluster number

In penalty term, the influence of K was smaller as bigger as K. It guarantees that a big number of clusters could survive in GA. The constant c determined the strength of influence of K. If smaller constant was used, than the influence was decreased also. By many experiments, the constant was fixed to 0.03. Pearson Correlation Coefficient was used as measurement of correlation elements and centroids. It was defined as

$$r = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{x_i - \mu_x}{\sigma_x}\right)\left(\frac{y_i - \mu_y}{\sigma_y}\right)$$

4) **Selection operation**: We used Roulette wheel operation to select solution candidates for next generation. In this time, elitism was applied to select finest solution candidates for transition from former generation to later generation.

5) **Crossover operation**: To make new solution candidates, the uniform crossover operation (with 0.5-probability) was used. By this operation, the search space could be expanded.

6) **Mutation operation**: To prevent fitness value from staying in local maximum, the mutation operation was applied. In our mutation operation, one bit in solution which was represented to 6-bit binary string was randomly reversed.

Extracting data sample using bootstrap was used to reduce fitness calculating time[7]. Re-sampling strategy was adopted to maintain unbiased estimator of sample. So, selection samples which were selected before was permitted. In our paper, four samples were selected. Proposed GA was performed in these samples, and taken the cluster number about their own sample. The final cluster number was determined by the cluster numbers of samples. The procedure of extracting samples and merging outputs is shown in Fig. 1. The means of outputs on samples was assigned optimal cluster number.

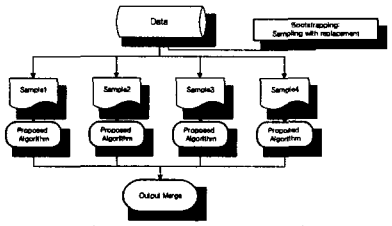

Fig. 1. Extraction samples and merging outputs

Above mentioned process of Clustering Using GA is shown in Fig. 2.

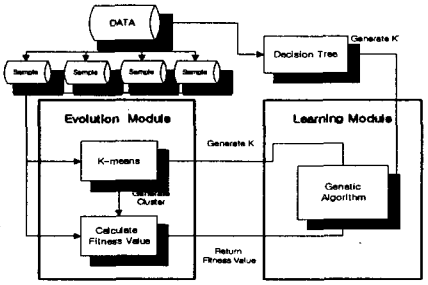

Fig. 2. Process of Clustering Using Genetic Algorithm

IV. EXPERIMENTAL RESULT

To verify the advantages of clustering with proposed algorithm, we selected two databases. These employed for experiments are obtained from UCI Machine Learning Repository and SAS Institute.

(1) Iris Plant database [13]
(2) Fish database [14]

To make initial population in GA, approximated cluster number K was determined using Decision Tree. The result of trained regression tree is shown in Fig. 3. This shows that terminal nodes from training Iris data are 2. So, the approximated cluster number

of initial population is 2. Like this procedure, the approximated cluster number of initial population for Fish data are 11.

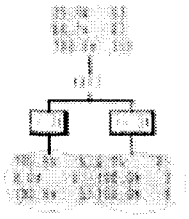

Fig. 3. Regression tree of Iris data

Table 2 shows approximated cluster numbers of these data.

Table 2. Approximated cluster numbers

	Iris	Arrhythmia
K	6	8

For our experiments, 4 samples of which size was thirty were extracted. Parameters for GA were set as table 3.

Table 3. Parameters for Genetic Algorithm

Crossover Rate	0.9
Mutation Rate	0.1
MaxIteration	30

Experiments were performed at twenty times for each data. The means and standard derivations of cluster numbers are shown in Table 4.

Table 4. Means and Standard derivation of cluster numbers

	Means	SD
Iris	3.19	0.5785
Fish	7.24	0.9783

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed hybrid genetic algorithm for K-means clustering. For the determination of initial population in GA, regression tree method was used. We used re-sampling of bootstrap for saving the computing time cost of fitness value calculation. In proposed algorithm, we thought the problem of optimal cluster size determination was searching process. By experimental results, our proposed algorithm was rapidly converged to optimal solution by searching of genetic algorithm. In future study, using other machine learning algorithm, the initial population is determined for genetic algorithm based clustering. And this clustering approach needs automatic processing by intelligent agents.

ACKNOWLEDGMENTS

This research was supported by BrainTech program sponsored by Korea Ministry of Science and Technology.

Reference

[1] M. Alabau, L. Idoumghar, R. Schott, New Hybrid Genetic Algorithms for the Frequency Assignment Problem, IEEE transactions on broadcasting, vol. 48, no.1, 2002.
[2] M. Baldonado, C. K. Chang, L. Gravano, A. Paepcke : The Stanford Digital Library Metadata Architecture, Int. J. Digit. Libr. 1, 1997.
[3] P. Christian, C. George, Monte Carlo Statistical Methods, Springer, 1999.
[4] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2001.
[5] J. W. Han, S. H. Jun, K. W. Oh, Cluster Merging Using Density based Fuzzy C-Means Algorithm, Proceedings of KFIS Spring Conference, May 3, 2003.
[6] S. H. Jun, J. Yang, K. W. Oh, Automatic detection of cluster size using machine learning algorithms, International Conference on Advances in Infrastructure for Electronic Business, Education, Science, and Medicine on the Internet, July 29- August 4, L'Aquila, Italy, 2002.
[7] T. M. Mitchell, Machine Learning, WCB McGraw Hill, 1997.
[8] M. Mitchell, An introduction to Genetic Algorithms, MIT Press, 1998.
[9] A. S. Pandya, R. B. Macy, Pattern Recognition with Neural Networks in C++, CRC Press, 1995.
[10] J. E. Park, S. H. Jun, K. W. Oh, Multi Intelligent Data Mining Agents for Automatic Clustering, KIISS2002, 2002.
[11] L. Rreiman, J. H. Friedman, R. A. Olshen, C. J. Stone, Classification and Regression Trees, Wadsworth International Group, 1984.
[12] M. Wall : GAlib, A C++ library of genetic algorithm components, in Manual, Mechanical Eng. Dept., MIT, 1996.
[13] http://www.ics.uci.edu/~mlearn /MLRepository.html
[14] http://www.sas.com