International Journal of Soft Computing, 1(4): 313-318, 2006

A Hybrid Genetic Algorithm and New Criterion for Determining the Number of Clusters

Sung-Hae Jun
Department of Bioinformatics and Statistics, Cheongju University,
Chungbuk, Korea

Abstract: The study of determining the number of clusters has had an effect on the performance of the clustering result. For example, in the K-means clustering algorithm, its clustering result is affected by the initial K as the number of clusters. But it has been determined by subjectively prior knowledge. Frequently this subjective determination may not be optimal. So, in this study, we proposed an objective method for determining the number of clusters using hybrid genetic algorithm. The initial population of our algorithm was generated by uniform distribution based on decision tree process. We also proposed a new criterion for evaluating the performance of clustering results. In the experiments, we verified our works using data sets from UCI machine learning repository.

Key words: Hybrid genetic algorithm, new clustering criterion, decision tree, the number of clusters

INTRODUCTION

The cluster is a collection of data objects. Clustering is the process of grouping the data clusters so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters[1]. That is, clustering algorithms attempt to optimize the placement of like objects into homogeneous classes or clusters[2]. Generally in the clustering, the number of clusters has been significantly considered for looking forward to good clustering results. But there are no completely satisfactory methods for determining the number of clusters for any type of clustering[3-5]. So, the number of clusters has been subjectively determined by the art of researchers. However, this approach was not only an inefficient approach but also an annoying problem in clustering[1,6,7]. Also the objective criteria have been needed for an efficient clustering.

The goal of our researches was to solve these problems in clustering. The proposed methods were a Hybrid Genetic Algorithm (HGA) based on decision tree and a Clustering Criterion based on Variance and Penalty (CCVP). HGA was a method for determining the number of clusters. And CCVP was a objective criterion for evaluating the clustering results.

Genetic Algorithm(GA) was a method of moving from one population of chromosomes to a new population using a kind of natural selection together with the genetics inspired operators of crossover, mutation and inversion[7]. GA was well suited for some of the most computational problems require searching through a population of possibilities for solutions. Biological evolution is a source of inspiration for addressing theses problems. Evolution is, in fact, a method of searching among an enormous number of prospects for solutions. The rules of evolution are remarkably simple, species evolve by means of random variation, followed by natural selection in which the fittest tend to survive and reproduce, thus propagating their genetic material to future generations[7,8]. GA is a widely applicable search technique that provides a global search for problems with many local optima. So, GA has been applied to many function optimization problems and are shown to be good in finding optimal and near optimal solutions[9]. We used GA to attempt to minimize the within cluster variance for optimal clustering. Determining the initial population is a very consequential component in GA. If the fittest individual is involved in the initial population, the solution searching time will be rapidly decreased. To make the initial population, two different methods are existed by random creating and specific information of specialized problem[10]. To make more efficient initial population, we proposed decision tree for GA. Using decision tree, we obtained an approximate number of clusters for the initial population. Namely it was used as seed of the initial population. The initial population was orderly created in and around this seed. GA is also a tool for optimization.

The clustering algorithm is to optimize the placement of like objects into homogeneous clusters. So GA can be an efficient tool for clustering[2,9,11-14]. In this study, based

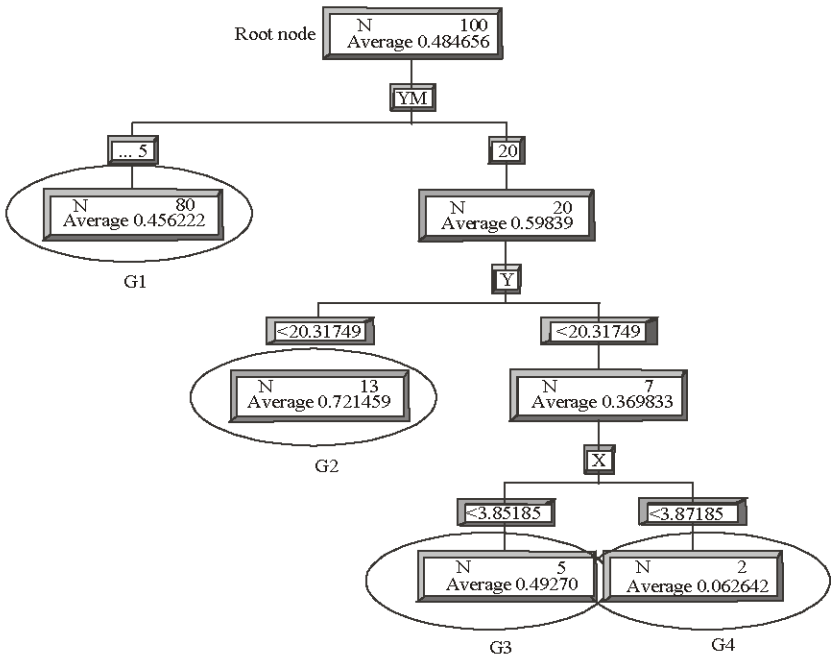

Fig. 1. An example of decision tree for HGA

on a fitness function for deciding the performance of cluster, our HGA converged to optimal number of clusters using repeated genetic operations w ith bootstrapping. And we checked the evaluation of clustering result usingproposed new criterion. In our experimental results using the data sets from UCI machine learning repository, we verified proposed methods[15].

A HYBRID GENETIC ALGORITHM

Initializing population using decision tree: The study of K-means was used in HGA as a clustering component. In solving clustering problem, traditional methods, for example, the K-means algorithm and its variants, usually ask the user to provide the number of clusters. Unfortunately, the number of clusters in general is unknown to the user. Therefore, the clustering becomes a tedious trial and error work and the clustering result is often not very promising especially when the number of cluster is large[8]. That is, sensitivity to initial points and convergence to local optima are usually among the problems affecting the K-means algorithm[16]. So, we proposed the approach of determining the initial population of GA using decision tree. Decision tree models can be built to solve either classification or regression problems, though they are most commonly and naturally used for classification[17]. This 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. The topmost node in a tree is the root node. In the decision tree, a majority voting is performed to assign a class label to the leaf while the mean of the objective attribute is computed and used as the predictive value[18]. An example of the decision tree for our HGA is shown in the following figure.

In Fig. 1, the number of total terminal nodes was 5. So the initial number of clusters for initial population was selected as 5. From this, we can construct initial population using the chromosome, (000101). This was an Study construct the initial population for determining the number of clusters. But we can not directly determine the number of clusters by the result of decision trees. Because it was focused not clustering but classification. Our regression tree model was split by minimizing the within node sum of squares. In other words, the total squared deviations of the actual value of the predicted variable with the average value within the node were used[17]. For deciding the target attribute of regression tree, we calculated the variances of all attributes[1]. The variance of attribute represents the fluctuation over given data. Because the bigger is the variance, the more is the importance of the attribute, we determined the attribute with the biggest variance to the target attribute. So, the type of decision trees was determined according to

Table 1: Determinded type of decision trees

	Target variable	
	Qualitative	Quantitative
Decision tree types	Classification tree	Regression tree
Used algorithms	Chi-square testing, Gini diversity index, Information gain ratio	Least square criteria, Total squared deviation

variable type of target. This is summarized in the following Table 1[18].

Determining the number of clusters: Our HGA for determining the number of clusters was comprised the followings.

Description of an individual: Each individual represented a possible solution to the problem and was composed of a chromosome. In HGA, each individual represented the number of clusters using 6-bit binary encoding. We used 6 bits string, because of it enough for representing the number of clusters. But it can be larger in study of need.

Construction of initial population: The initial number of clusters was taken by decision tree approach. It was set as the initial population in HGA. After decision tree process, the number of terminal nodes was used for constructing the initial population of HGA. Using this number, we decide upper value of uniform distribution[19]]. So, we selected the candidate solutions from uniform distribution for constructing the initial population. The uniform distribution for HGA is shown in the following.

$$CS_j \sim U(1, u_0), \quad j = 1, 2, \cdots, N_{cs} \tag{1}$$

where, CS_j is the *j*th candidate solution and u_0 is the upper value of uniform distribution. The population size is determined by N_{cs}. In HGA, the lower value of uniform distribution is decided by 1 because the smallest number of clusters is 1.

Fitness function: The fitness function of HGA was defined as the following.

$$\text{Fitness} = \begin{cases} 0 & \text{if } M = 0 \\ \dfrac{1}{e^{-c/M} \sum_{i=1}^{n} d(o_i, C_i)} & \text{if o.w.} \end{cases} \tag{2}$$

where o_i is the ith object in data set and C_i is the centroid of cluster that holds ith object. M represents the number of clusters. c is a constant value, heuristically set to 0.03. $d(o_i, C_i)$ is Euclidean distance between o_i and C_i[18]. This function is composed of two parts which are $\sum_{i=1}^{n} d(o_i, C_i)$ and $e^{-\frac{c}{M}} \cdot \sum_{i=1}^{n} d(o_i, C_i)$ represents the dissimilarity between o_i and C_i. The penalty of excessively increasing the number of c lusters was expressed by $e^{-\frac{c}{M}}$. In penalty term, the value of fitness function was decreased by increasing M. And c determined the strength of influence of M. The small c can decrease the effect of M.

Selection operator: We used roulette wheel operation to select solution candidates for next generation. At this time, the elitism scheme was applied to select the finest solution candidates for transition from former generation to later generation.

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

Mutation operator: To prevent fitness value from staying in local maxima, the mutation operator was applied. In our mutation, one bit in individual which was represented to 6-bit binary encoding was randomly reversed.

K-means operator: The GA minimizes the Euclidean distance between a data point and the centre of cluster. We defined the K-means operator as the following.

$$D = \sum_{m=1}^{M} \sum_{x \in G_m} (x - R_m)^2 \tag{3}$$

where M is the number of clusters and R_m is the centre of cluster G_m.

To conclude, proposed HGA is shown in the following pseudo code.

Hybrid Genetic Algorithm

Input:
Mutation probability, P_m;
Crossover rate, *CrossRate*;
Maximum number of iteration, *MaxIteration*;

Output:
Optimal number of clusters, *M*;
K-means clustering;

Begin
Generate population P_0;
Performing the decision tree
- determine the target variable;
- if (data type of target variable = qualitative) then use classification tree model; else use regression tree model;
- Run decision tree process;
- Count the number of terminal nodes, T_{num};

.Determining initial population for the number of clusters using the result of decision trees
- set $u_0 = T_{num}$;
- using $U(1, u_0)$;

Evaluate P_0;
For 1 to MaxIteration do {
    Select two parents p1 and p2 from P_{i-1};
    Offspring <- (p1, p2);
    Mutate offspring;
    Evaluate offspring and assign it to fitness;
    Add offspring to P_i;
}
Return the best value, *M* from $P_{MaxIteration}$;
K-means clustering using *M*;
End

Additionally we used re-sampling strategy to maintain unbiased samples. Our re-sampling method was based on bootstrap[20,21].

New Criterion for clustering: A good results of clustering have high intra-cluster similarity and low inter-cluster similarity[7]. And we proposed a new criterion for evaluating the results of clustering on the ground of above viewpoints. This criterion was composed of two parts which were the variance of objects in clusters and the penalty of excessive increasing the number of clusters. We called it Clustering Criterion based on Variance and Penalty (CCVP). Its measure was defined as the following.

$$CCVP_M = \frac{1}{M}\sum_{i=1}^{M}\bar{v}_i + \frac{1}{\bar{V}_M}M \quad (4)$$

In the above equation, M was the number of clusters and $\bar{v}_i$ was the average of variances of objects in the *i*th cluster. $\bar{V}_M$ was the variance of M clusters. This was defined as the following.

$$\bar{V}_M = \frac{1}{M-1}\sum_{j=1}^{M}(c_j - \bar{c})^2 \quad (5)$$

In the eq. (3), c_j was the center of jth cluster and $\bar{c}$ was the average of the centers of M clusters. The smaller the CCVP value was, the better the clustering result was.

RESULTS AND DISCUSSION

The Iris Plants, Cardiac Arrhythmia and Glass Identification data sets from UCI machine learning repository were used for our experiments[15]. The following Table shows the summary of the data sets. The Iris data set is simple because it has 4 attributes and 50 instances per each class equally. Also we considered the Arrhythmia data set as complicated data because it has 279 attributes, 452 instances and 16 classes. But, in Arrhythmia data set, 8 attributes have 0 or few as the number of instances. So, we removed these 8 attributes from Arrhythmia data. The Glass data set was used as a moderate data. It was not simple, also not complicated. Therefore, the descriptive specifics of the experimental data sets are shown in the following Table 2.

In the next, for determining initial population in HGA, we used classification tree method because all target variables of 3 data sets were qualitative variables. According to classification tree method, we found that the numbers of terminal nodes were 6, 8 and 12 for Iris, Arrhythmia and Glass. Table 3 shows the results of our initial study.

After decision tree process of Iris data set, the number of terminal nodes was 6. It was used for upper value of uniform distribution. So, the candidate solutions for making initial population were selected from U(1.6).

Table 2: Summarization of data sets

Numbers of items	Iris	Arrhythmia	Glass
# of instances	150	429	214
# of attributes	4	279	9
# of classes(labels)	3	8	7

Table 3: Summarization of data sets

Data sets	# of terminal nodes	Upper value of uniform distribution (u_0)	Selective distribution for initial population
Iris	6	6	U (1, 6)
Arrhythmia	8	8	U (1, 8)
Glass	12	12	U (1, 12)

Table 4: Parameters set up for HGA

Crossover rate	0.9
Mutation rate	0.1
Maximum iteration	30

Table 5: Mean and standard deviation of the number of clusters

Data sets	Mean	S.D.
Iris	3.35	0.1210
Arrhythmia	8.42	0.9944
Glass	7.19	1.4522

Table 6: CCVP values according to # of clusters

Iris		Arrhythmia		Glass	
# of Clusters	CCVP	# of Clusters	CCVP	# of Clusters	CCVP
2	0.70	7	1.98	6	1.55
3	0.64	8	1.73	7	1.23
4	0.69	9	2.33	8	1.69

Table 7: The number of clusters and VC values

	Iris (# of clusters=3)		Arrhythmia (# of clusters=8)		Glass (# of clusters=7)	
Clustering methods	Accuracy rate (%)	CCVP	Accuracy rate (%)	CCVP	Accuracy rate (%)	CCVP
HGA	98.67	0.64	94.17	1.73	96.26	1.23
Hierarchical clustering	80.67	1.84	87.65	2.21	85.98	2.01
K-means algorithm	93.33	1.11	90.68	2.18	89.25	1.57
SOM	88.00	1.46	82.75	2.91	80.84	2.38

The study of Arrhythmia and Glass data sets were the same the study of Iris data. For our experiments, 4 samples were extracted by re-sampling. Each sample size was 30. The parameters for HGA were set up as the following Table 4.

They were determined heuristically. Our experiments were performed at twenty times for each data because we needed the values of mean and variance of the number of clusters. According to HGA, we got the result of optimal number of clusters. Table 5 shows the means and standard deviations of optimal number of clusters.

We determined that the number of clusters for the Iris data was 3 because the nearest integer of 3.35 was 3. In the same way, the numbers of clusters for Arrhythmia and Glass were decided to 8 and 7. Also we found that the dispersion of the numbers of clusters was stable because the values of standard deviation of all data sets were small.

CCVP values of our new criterion for optimal clustering were shown in the following Table 6.

Using (2), we determined that the number of clusters for Iris data was 3 because it had the smallest CCVP value. Similarly, the numbers of clusters for Arrhythmia and Glass data sets were determined by 8 and 7. Lastly we compared our HGA with other clustering algorithms which were hierarchical clustering in statistics, K-means clustering algorithm and Self Organizing Maps (SOM) by the CCVP measure[22,23]. Also proposed CCVP criterion was compared with accuracy rate as generally objective measure because the clustering problem is defined as the problem of classifying n objects into M clusters without a priori knowledge[2,9,13,24,25].

In the Iris data set, the CCVP value of HGA was the smallest of the comparative methods. In the studys of Arrhythmia and Glass data sets, the performances of HGA were identical with the result of Iris data. For more objective evaluation, we used the accuracy rate as the popular measure of clustering. It has been used in many clustering researches. According to the experimental result by the accuracy rate, we found that the accuracy rate of HGA was the highest of the comparative methods in all data sets. So, we verified the improved performance of HGA and the validity of CCVP criterion.

CONCLUSION

In this study, we proposed the HGA which was GA based on decision tree for determining the number of clusters. For the determination of the initial population in GA, we used the decision tree method. Also CCVP, a new measure for evaluating the performance of clustering was proposed. In HGA, we thought the problem which determined the optimal number of clusters was searching process using GA. CCVP was consisted of the variance of objects within cluster and among clusters and the penalty of increasing the number of clusters. Therefore, our methods can automatically decide the number of clusters and do the clustering work. By experimental results, we found that the optimal number of clusters was efficiently determined by HGA and CCVP. In future studies, we will compare the HGA with other unsupervised learning algorithms on the computing time. Also the co-evolutionary computing approach for HGA will be considered.

REFERENCES

1. Han, J. and M. Kamber, 2001. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers.
2. Bezdek, J.C., S. Boggavarapu, L.O. Hall and A. Bensaid, 1994. Genetic algorithm guided clustering. IEEE World Congress on Computational Intelligence, pp: 34-39.
3. Bock, H.H., 1985. On Some Significance Tests in Cluster Analysis, J. Classification, pp: 77-108.
4. Everitt, B.S., 1979. Unresolved Problems in Cluster Analysis, Biometrics, pp: 169-181.
5. Hartigan, J.A., 1985. Statistical Theory in Clustering, Journal of Classification, pp: 63-76.
6. Mitchell, T.M., 1997. Machine Learning. WCB McGraw Hill.
7. Mitchell, T.M., 1998. An introduction to Genetic Algorithms. The MIT Press.
8. Wall, M., 1996. GAlib: A C++ library of genetic algorithm components, in Manual: Mechanical Eng. Dept., MIT.
9. Krishna, K. and Narasimha M. Murty, 1999. Genetic K-means algorithm. IEEE Transactions on Systems, Man and Cybernetics, Part B, pp: 433-439.

10. Baldonado, M., C.K. Chang, L. Gravano and A. Paepcke, 1997. The Stanford Digital Library Metadata Architecture. Intl. J. Digit. Libr. 1.
11. Bandyopadhyay, S. and U. Maulik, 2001. Nonparametric genetic clustering: comparison of validity indices. IEEE Transactions on Systems, Man and Cybernetics, Part C, pp: 120-125.
12. Buckles, B.P., F.E. Petry, D. Prabhu, R. George and R. Srikanth, 1994. Fuzzy clustering with genetic search, IEEE World Congress on Computational Intelligence, pp: 46-50.
13. Douzono, H., S. Hara, S. Kawamoto and Y. Noguchi, 1998. A Clustering Algorithm Using Genetic Algorithm with Competitive Individuals, Second International Conference on Knowledge-Based Intelligent Electronic Systems, pp: 491-496.
14. Lin, Y.T. and Shiueng, B.Y., 1997. Genetic algorithms for clustering, feature selection and classification. Intl. Conference on Neural Networks, pp: 1612-1616.
15. UCI machine learning repository, http://www.ics.uci.edu/~mlearn/MLRepository.html
16. Bradley, P.S. and U.M. Fayyad, 1998. Refining Initial Points for K-Means Clustering, The fifteenth Intl. Conference on Machine Learning.
17. TeraMiner Development Team, 2000. Machine Learning. Technical White Paper, rev. 1.
18. Breiman, L., J.H. Friedman, R.A. Olshen and C.J. Stone, 1984. Classification and Regression Trees. Wadsworth, Belmont.
19. Casella, G. and R.L. Berger, 1990. Statistical Inference, Duxbury Press.
20. Christian, P. and C. George, 1999. Monte Carlo Statistical Methods, Springer.
21. Dong, Y., Y. Zhang and C. Chang, 2004. Multistage random sampling genetic-algorithm-based fuzzy c-means clustering algorithm. Proceeding of International Conference on Machine Learning and Cybernetics, pp: 26-29.
22. Hastie, T., R. Tibshirani, J. Friedman, 2001.The Elements of Statistical Learning. Springer.
23. Kohonen, T., 1997. Self-Organizing Maps. Springer.
24. Fred, A.L.N. and A.K. Jain, 2003. Robust Data Clustering, IEEE Computer Society Conference on Computer Vision or Pattern Recognition, pp: 128-133.
25. Tseng, L.Y. and S.B. Yang, 1997. Genetic Algorithms for Clustering, Feature Selection and Classification, Intl. Conference on Neural Networks, pp: 1612-1616.