On-line Rule Updating System Using Evolutionary Computation for Managing Distributed Database

Wirarama Wedashwara, Shingo Mabu, Masanao Obayashi and Takashi Kuremoto
Graduate School of Science and Engineering, Yamaguchi University
Tokiwadai 2-16-1, Ube, Yamaguchi 755-8611, Japan
E-mail:{ t001we, mabu,m.obayas, wu}@yamaguchi-u.ac.jp

Abstract

This research proposes a decision support system of database cluster optimization using genetic network programming (GNP) with on-line rule based clustering. GNP optimizes cluster quality by reanalyzing weak points of each cluster and maintaining rules stored in each cluster. The maintenance of rules includes: 1) adding new relevant rules; 2) moving rules between clusters; and 3) removing irrelevant rules. The simulations focus on optimizing cluster quality response against several unbalanced data growth to the data-set that is working with storage rules. The simulation results of the proposed method show its priority comparing to GNP rule based clustering without on-line optimization.

Keywords: Genetic Network Programming, Rule Based Clustering, Cluster Optimization

1. Introduction

Nowadays many large scale database systems with very high data growth are being utilized to improve the global human activity, such as communication, social networking, transaction, banking, etc. A distributed database management system becomes one of the solutions to improve data access speed by organizing data in multiple storages for multiple types of user accesses. Problems of distributed database management systems are not only how to manage the large number of data, but also how to organize data patterns in the distributed storages. Clever data organization is one of the best ways to improve the retrieval speed and reduce the number of disk I/Os and thereby reduce the query response time.

In this paper, we propose a decision support system for database cluster optimization using Genetic Network Programming (GNP) with on-line rule based clustering. In the proposed system, an on-line algorithm is utilized to maintain the cluster adaptability against several unbalanced data growth. For example, the unbalanced data growth occurs when different kinds of items (data) comparing to the items stored in the current database begin to be stored as the time goes on (the trend of data is changed).

This paper is organized as follows. Section 2 describes a management of distributed database, section 3 describes the on-line rule updating system, section 4 shows the simulation results, and finally section 5 is devoted to conclusions.

2. Management of Distributed Database

2.1. *Rule Based Clustering*

Rule based clustering is one of the solutions to provide automatic database clustering and interpretation of data storage patterns. Rule based clustering represents data

Published by Atlantis Press

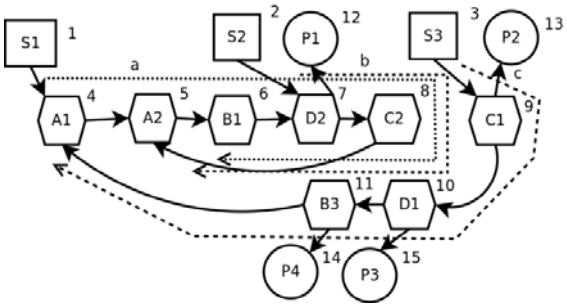

Fig. 1. GNP Implementation on Cluster Optimization.

Table 1. Gene Structure of GNP Corresponding to the Program in Fig. 1.

i	NTi	Ai	Ri	Ci
1	1	0	0	4
2	1	0	0	7
3	1	0	0	9
4	2	A	1	5
5	2	A	2	6
6	2	B	1	7
7	2	D	2	8,12
8	2	C	2	5
9	2	C	1	10,13
10	2	D	1	11,15
11	2	B	3	4,14
12	3	1	1	0
13	3	2	1	0
14	3	3	2	0
15	3	4	2	0

patterns as rules by analyzing database structures on both of attributes and records[3,4]. Each clusters partitioned with different rule pool which each rule have a similarity to rules in internal cluster and dissimilarity to rules in external cluster.

2.2. *Genetic Network Programming*

Genetic network programming (GNP), an evolutionary optimization technique with directed graph structures, is used for data classification. GNP has a distinguished representation ability with compact programs, which is derived from the re-usability of nodes that is inherently equipped function of the graph structures. For the purpose of rule based clustering, GNP is useful to handle rule extraction from data-sets by analyzing the records.

A graph structure of GNP consists of three kinds of nodes: start nodes, judgment nodes and processing nodes. Start nodes represent the start positions of the node transition; judgment nodes represent attributes to be examined in a database; and processing nodes represent the cluster numbers to which rules are assigned. The node preparation for GNP rule extraction contains two phases: node definition and node arrangement. The purpose of node definition is to preparing judgment nodes that will be combined to create rules, and node arrangement is to select important nodes for efficiently extracting a large number of rules. The node arrangement is executed by the following two sequential processes. The first process is to find template rules and the second process is to generate rules combining templates created in the first process. Templates are extracted to obtain combinations of attributes that frequently happen in the data-set. In the template extraction process, only a few numbers of attributes are used in GNP rule extraction. It aims to increase the possibility to get templates with high support*. This node arrangement method has shown better clustering results compared to the method without template rules (i.e., all the attributes are simultaneously used in the rule extraction process)[9]. The rule generation in both processes is carried out by evolution (crossover and mutation) of the graph structures[1, 2].

2.3. *Silhouette*

Silhouette value is used to evaluate the clustering results. Silhouette provides a succinct graphical representation of how well each object lies v (1) s cluster[5, 8]. Silhouette value is calculated by Eq. ( )

$$s = \frac{b-a}{\max\{a,b\}} = \begin{cases} 1-a/b & if\ a<b \\ 0 & if\ a=b \\ b/a-1 & if\ a>b \end{cases}$$

* Support shows the occurrence frequency of the rule in the database.

Table 2. Example of Rule Extraction for Cluster Optimization.

Start Node	Extracted Rules	Support	Processing Node	Optimization
1	A_1^B_1	3	*P1*	Add rule to cluster 1
	A_1^B_1^D_2	1	*P1*	Add rule to cluster 1
	A_1^B_1^D_2^C_2	1	*P1*	Add rule to cluster 1
2	D_2^C_2	2	*P2*	Add rule to cluster 2
	D_2^C_2^A_2	1	*P2*	Add rule to cluster 2
	D_2^C_2^A_2^B_1	0	*P2*	Add rule to cluster 2
3	C_1^D_1	2	*P3*	Remove rule from cluster 3
	C_1^D_1^B_3	1	*P4*	Remove rule from cluster 4
	C_1^D_1^B_3^A_1	1	*P1*	Add rule to cluster 1

s: Silhouette value for a single sample. The Silhouette value for a set of samples is given as the mean of the Silhouette values of each sample, *a*: mean distance between a sample and all the other points in the same cluster, *b*: mean distance between a sample and all the other points in the second nearest cluster. A good clustering result will make the silhouette value be close to 1.0 and a bad clustering result will make it be close to -1.0. Silhouette value is used as a threshold (condition) of the rule updating.

3. An On-line Rule Updating System

GNP is used to extract rules from a data-set by analyzing all the records. Phenotype and genotype structures of GNP are described in Fig. 1 and Table 1, respectively. In Fig. 1, each node has its own node number *i* (1–15), and in Table 1, the node information of each node number is described. The program size depends on the number of nodes, which affects the amount of rules created by the program. 1) Start nodes (rectangle) represent the start points of the sequences of judgment nodes which are executed sequentially according to their connections. Multiple placements of start nodes will allow one individual to extract a variety of rules. 2) Judgment nodes (hexagon) represent attributes of the data-set which are represented by *Ai* (in Table 1) showing an index of attribute such as price, stock, etc., and *Ri* showing a range index of attribute *Ai*. For example, *Ai=A* represents price attribute, and *Ri=1* represents value range [0,50] and *Ri=2* represents value range [51,80]. 3) Processing nodes (round) show the end points of the sequences of judgment nodes and processes the rule updating in a cluster whose cluster number is described as *Ai* in the processing node . *Ri* shows the function of the rule updating, that is, *Ri* =1 means adding the rule, and *Ri* =2 means removing the rule. For example, *P1* in Fig. 1 (node number *i*=12 in Table 1) processes an addition of extracted rules to cluster number 1. Multiple placement of processing nodes will make one individual extract variety of rules. The sequences of nodes starting from each start node (S_1,S_2,S_3) are represented by dotted lines *a*, *b* and *c*. A node sequence flows until support for the next combination of the judgment node (attribute) does not satisfy the threshold. In the node sequence, if the nodes with the attributes that have already appeared in the previous node sequence appear, the nodes will be skipped. Candidate rules extracted by the program of Fig. 1 are shown in Table 2.

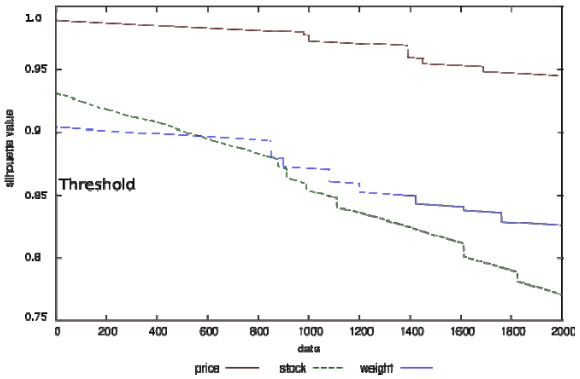

Fig. 2. Example of Silhouette Values

When updating rules in each cluster, it is important to find attributes that are not matched with the latest data. Therefore, the attributes to be considered in the rule updating are determined by the following procedure. Fig. 3 shows an example of the Silhouette values when Silhouette values are calculated between each attribute and data in a cluster. In Fig. 2, threshold is set at 0.85, and only the attributes with Silhouette values under 0.85

will be selected for the attributes of the judgment nodes in the rule updating process. In Fig. 2, the attribute of price is not included in the rule updating of GNP because its silhouette values are never under 0.85 (rule updating is not necessary), however, the values of stock and weight for some data are under 0.85, thus those attributes are included in GNP for updating rules. Each cluster has dominant values of attributes which are the anchors of cluster quality, which then influence the silhouette values. For example, when 10 [kg] is a dominant value for the attribute of weight in a cluster, larger values than 10 [kg] would have a lower silhouette and should be moved to another cluster to improve the cluster quality.

In summary, the proposed method consists of two processes:

1) Main rule extraction process, which is a standard rule extraction with GNP considering all the attributes in a data-set. This process is executed to make initial clusters for the initial data-set;

2) Rule updating process, which is executed for only the data that have lower silhouette values than the threshold. This process is repeated until good average value of silhouette (cluster quality) is obtained.

The flowchart of the above processes is shown in Fig. 3.

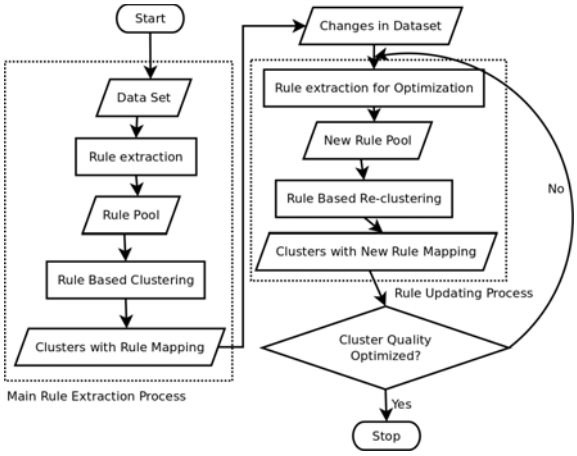

Fig. 3. Flowchart of Proposed Method

4. Simulation

Two kinds of simulations were carried out:

Simulation I: Comparison of silhouette values between different rule updating frequencies;

Simulation II: Comparison of silhouette values and iterations between different setting of thresholds.

4.1. *Simulation Environment*

The initial data-set used in the simulations contains 1000 data with eight attributes. The data-set is created by randomly determining the attribute values in the fixed ranges of each attribute. For example, attribute 1 has an integer value between 1 and 10, while attribute 2 has a value between 1000 and 2000. To evaluate the adaptability of the proposed method, the number of data will be increased by adding randomly generated data or decreased by deleting data selected randomly.

4.2. *Comparison of Silhouette Values between Different Rule Updating Frequencies*

The first simulation focuses on verifying the cluster adaptability against several unbalanced data growth of the data-set, where cluster adaptability is evaluated by silhouette values.

The adaptability is evaluated in terms of the following two points:

1) The number of data in the data-set is changing as the time goes on. The initial number of data is 1000, then every time step, 1000 new data is added to the data-set. After the number of data reaches 6000, 1000 data is decreased every time step;

2) The rule updating is executed every after the predefined number of new data are given to the data-set (the predefined number is called "rule updating frequency"). For example, if the rule updating frequency is 1000, the rule updating is executed every increments or decrements of 1000 data.

The comparisons of the simulation results were carried out between four settings, i.e., the proposed method with rule updating frequency of 1000, 2000 and 4000, and the clustering method of standard GNP without on-line rule updating.

The silhouette values obtained by the four methods are shown in Table 3. Star marks (*) on the side of silhouette values indicate the times when the rule updating is carried out. In the case of the rule updating frequency of 4000, the increment of silhouette values in step 5 and 9 can be observed, which means that the rule updating is effectively carried out. The best results are obtained by the rule updating frequency of 1000, where silhouette values are stable with relatively high level compared to other frequency parameters. In step 3, although the rule updating is carried out by rule update frequency 1000 and 2000, the silhouette values decreases, because the degree of the data change is relatively large in this step.

Table 3. Simulation Result of Rule Updating Frequency and Number of Data Comparison.

Step	Number of Data	Increment/Decrement	Silhouette values			
			No rule updating	frequency: 1000	frequency: 2000	frequency: 4000
1	1000 (Default)	-	0.967	0.967	0.967	0.967
2	2000	+1000	0.945	0.965*	0.944	0.947
3	3000	+1000	0.902	0.923*	0.915*	0.899
4	4000	+1000	0.892	0.935*	0.902	0.895
5	5000	+1000	0.882	0.912*	0.909*	0.903*
6	6000	+1000	0.812	0.902*	0.897	0.887
7	5000	-1000	0.787	0.892*	0.901*	0.821
8	4000	-1000	0.765	0.901*	0.888	0.797
9	3000	-1000	0.723	0.912*	0.892*	0.815*
10	2000	-1000	0.698	0.909*	0.879	0.802
Average			0.832	0.938	0.923	0.885

Table 4. Comparison of Silhouette Values and Iteration between Different Setting of Threshold.

Step	Number of Data	Increment	Number of iteration for each threshold of Silhouette value							
			1.0		0.85		0.7		0.5	
			Sil	Iter	Sil	Iter	Sil	Iter	Sil	Iter
1	1000 (Default)	-	0.971	153	0.971	153	0.971	153	0.971	153
2	2000	+1000	0.945	165	0.935	53	0.923	45	0.912	15
3	3000	+1000	0.902	201	0.913	65	0.902	53	0.895	8
4	4000	+1000	0.905	265	0.907	89	0.873	75	0.846	11
5	5000	+1000	0.902	354	0.913	102	0.851	89	0.802	9
6	6000	+1000	0.878	402	0.837	112	0.898	99	0.816	12
Average			0.924	278	0.905	133	0.935	126	0.893	83

Sil: Silhouette value, Iter: iteration

4.3. *Iterations between Different Setting of Thresholds*

The second simulation focuses on comparing the iteration time and silhouette values with different settings of thresholds for several unbalanced data growth of the data-set. Iteration time in this simulation means the number of individuals generated to cover all the data in the evolution for the rule updating. Lower iteration time is required to minimize hardware resource usage, so on-line processing in this paper means that the re-organizing the clusters can be executed in less iteration time than the method without on-line rule updating. Data-set used in the simulations has 1000 data with 8 attributes. The adaptability is evaluated in terms of the following two points. 1) The number of data in the database is changed as the time goes on. The initial number of data is 1000, then every time step, 1000 new data is added to the database. 2) Several settings of thresholds for selecting attributes are analyzed. In this case, the silhouette values of each attribute with the current rules in the cluster is calculated, and if the silhouette values are lower than the minimum value, i.e., a threshold, the attributes showing the lower silhouette values are included in the rule updating process. For example, if the threshold is 0.5, the attributes with silhouette values being lower than 0.5 will be added to the rule updating process. Higher threshold increases the number of attributes to be reanalyzed in the rule updating process, which would increase the iteration times. The comparisons of the simulation results are carried out between four settings of thresholds, i.e., the proposed method with the threshold of 0.5, 0.7, 0.85, and 1.0. The threshold 1.0 means that all attributes will be added to the rule updating process.

The silhouette values and iteration times obtained by the four settings of thresholds are shown in Table. 4.

The setting of 0.7 shows the best average silhouette that is slightly better than the higher threshold of 1.0 and 0.85. This result shows that the higher thresholds do not always have better re-clustering results. This kind of situations are caused when more attributes are contained in the rule updating process, that is, the possibility to ruin the placement of data, that have been already optimized in the clusters, would increase. Threshold setting of 0.5 has the lowest average silhouette value. This is because the small number of attributes contained in the rule updating process also does not sufficiently optimize the cluster quality. On the other hand, in the comparison of the iteration time, the lowest threshold setting of 0.5 results in the lowest iteration time, and the highest threshold setting of 1.0 results in the highest iteration time. Higher thresholds tend to include more attributes in the rule updating, which will require more iteration times to process many attributes. So when we use the proposed on-line clustering mechanism, the balance between the cluster quality (silhouette values) and the iteration times need to be determined appropriately.

5. Conclusions

This paper proposed a new on-line rule updating system for distributed database with unbalanced data growth. The simulation results of the proposed method showed the better clustering results and iteration time comparing to GNP rule-based clustering without on-line adaptation. In the future, we will apply fuzzy membership functions to attribute judgment to make rules with better clustering ability.

References

1. Kaoru Shimada, Kotaro Hirasawa, and Jinglu Hu. Genetic network programming with acquisition mechanisms of association rules, *JACIII.* 10(1) (2006) 102‐111.
2. Shingo Mabu, Ci Chen, Nannan Lu, Kaoru Shimada, and Kotaro Hirasawa. An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming Systems, *Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on.* 41(1) (2011) 130‐139.
3. Mansoori, E.G., FRBC: A Fuzzy Rule-Based Clustering Algorithm, *Fuzzy Systems, IEEE Transactions.* 19(5) (2011) 960-971.
4. Sinaee, M.; Mansoori, E.G., Fuzzy Rule Based Clustering for Gene Expression Data, *Intelligent Systems Modelling & Simulation (ISMS)*, 4th International Conference (2013) 7-11.
5. Zhao, JiangFei and Huang, TingLei and Pang, Fei and Liu, YuanJie, Genetic algorithm based on greedy strategy in the 0-1 knapsack problem, *Genetic and Evolutionary Computing*, 3rd International Conference on (2009) 105-107.
6. Sylvain Guinepain and Le Gruenwald. Using cluster computing to support automatic and dynamic database clustering, *Cluster Computing IEEE International Conference*, (2008) 394‐401.
7. Sylvain Guinepain and Le Gruenwald. Automatic database clustering using data mining. Database and Expert Systems Applications, DEXA IEEE 06(17) (2006) 124‐128.
8. Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, and Junjie Wu. Understanding of internal clustering validation measures, *Data Mining (ICDM) IEEE 10th International Conference*, 10 (2010) 911‐916.
9. Wirarama Wedashwara, Shingo Mabu, Masanao Obayashi and Takashi Kuremoto. Implementation of Genetic Network Programming and Knapsack Problem for Record Clustering on Distributed Database, *SICE Annual Conference (SICE), 2014 Proceedings of the. IEEE*, (2014) 935-940.