Evolving Associative Classifier for Incomplete Database Using Genetic Network Programming

Kaoru Shimada
(1) Fukuoka Industry, Science & Technology Foundation, Japan
(2) IPS Research Center, Waseda University
k.shimada@ruri.waseda.jp

Kotaro Hirasawa
Graduate School of Information, Production and Systems, Waseda University
Wakamatsu, Kita-Kyushu, 808-0135, Japan
hirasawa@waseda.jp

ABSTRACT

An evolving classification method for incomplete database has been proposed as an extension of Genetic Network programming (GNP) based rule extraction. An incomplete database includes missing values, however, the method can extract class association rules and build a classifier. The proposed method evolves the classifier using the labeled instances by itself as acquired information. We have evaluated the performance of the proposed method using artificial incomplete data set. The results showed that the proposed method has a potential of gathering useful information for classification through its evolutionary process.

Categories and Subject Descriptors

H.2.8 [Database Applications]: Data mining

General Terms

Algorithms

Keywords

Classification, Association Rule, Genetic Network Programming

1. INTRODUCTION

Recently, association rule mining tools for incomplete data set have been proposed using Genetic Network Programming (GNP) [1]. An incomplete database includes missing data in some instances. GNP is one of the evolutionary optimization techniques, which uses the directed graph structures as genes. In the GNP based rule extraction method, rules satisfying the given conditions are stored in a rule pool through GNP generations. GNP evolves in order to store new interesting rules into the pool as many as possible, not to obtain the individual with highest fitness value.

In this paper, we propose a new method for building an evolving classifier which can gather useful information itself to expand the target data for classification through evolutionary process. In the GNP based rule extraction method, we can quit the rule extraction anytime. Applying extracted rules at the moment for classification, we can obtain new labeled instances. If we join these just labeled data into training data, then extended training data can be constructed. We can repeat this operation and evolve the classifier using acquired data which labeled by itself.

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

2. EVOLVING CLASSIFIER USING GNP

Let A_i be an attribute in the database and its value be 1 or 0, and C be the class label. Suppose that class label is $C=1$ or $C=0$. X denotes the combinations of attributes like $X=(A_j=1)\wedge\cdots\wedge(A_k=1)$. In this paper, we define the important Class Association Rules (CARs) as satisfying the following:

$$support(X\rightarrow(C=k))\geq sup_{min}, \quad (1)$$

$$\chi^2(X\rightarrow(C=k))>\chi^2_{min}, \quad (2)$$

$$confidence(X\rightarrow(C=k))\geq conf_{min}, \quad (3)$$

where, sup_{min}, χ^2_{min} and $conf_{min}$ are the threshold values given by users [1]. The method described in [2] for building a classifier using GNP-based rule extraction is extended for the case of the incomplete data set as follows:
[Input] A set of CARs and an instance to be classified
[Output] Class predicted by the classifier
[Method]
1) $available(k)$: compute the total number of available rules satisfying $C=k$ in the classifier ($k=0,1$), where, the available rule is defined as the rule which can judge whether the instance satisfies the antecedent of the rule or not [1].
2) $match(k)$: compute the number of rules in the classifier, whose antecedent match the instance and satisfy $C=k$
3) $score(k)=\frac{match(k)}{available(k)}$
If $available(k)=0$ then $score(k)=0$
4) the instance is predicted in such a way that it belongs to the class having the highest $score$.

Fig. 1 shows the flow of the proposed evolving classifier. We repeat a cycle of rule extraction and classification until given finish condition. One cycle for classification is defined as *round* for a concept of upper layer of evolutionary process. In the proposed method, data are divided into three categories as follows:

- Division A: Set of the seed instances, that is, training data for the first building a classifier. All the instances are labeled in advance and fixed through generations.
- Division B: Set of labeled instances by the evolving classifier. At the initial generation, this division is empty.
- Division C: Set of unlabeled instances.

GNP based rule extraction for each class is done using all the instances in Division A and B. If the number of extracted rules are enough for given condition or spent a given number of GNP generation, then stop rule mining and build a clas-

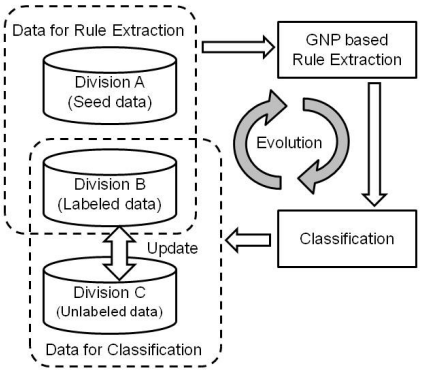

Figure 1: Flow of the proposed evolving classifier

sifier. The instances in Division B and C are tried to label based on the method described above. In order to obtain good labeling with confidence, we use the strict additional condition $max\{score(k)\} \geq score_{min}$, where, $score_{min}$ is the threshold value. If an instance in Division B is not labeled, then the instance moves to Division C. At the end of evolution, the gathered data in Division B brings us discovered information. In the case of some data are left in Division C, these data can be candidates of unknown or abnormal case. After the classification, we empty the rule pool in order to extract rules for next classification.

3. EXPERIMENTAL RESULTS

We used the same database named SNP_{com} in [1]. SNP_{com} has 100 attributes and 270 instances. The original data is The Mapping 500K HapMap Genotype Data Set (Affimetrix) [1]. The data has 4 class labels: CEU (90 instances), YRI (90), JPT (45) and CHB (45). We defined as $C=1$ in the case of CEU and $C=0$ in the case of YRI, respectively. Instances were divided for the initial data randomly and modified using artificial missing values as follows.

- Division A: 30 instances for $C=1$, 30 instances for $C=0$ (All the attribute values of A_{51}-A_{100} are missed)
- Division C: 30 instances for $C=1$, 30 instances for $C=0$ (All the attribute values of A_1-A_{25} and A_{76}-A_{100} are missed)

Division C: 30 instances for $C=1$, 30 instances for $C=0$ (All the attribute values of A_1-A_{50} are missed)
Class labels of instances in Division C are used for evaluation of the proposed method.

Settings for GNP was the same described in [1]. The condition of termination of GNP is 200 generations. We used $sup_{min}=0.03$ for (1), $\chi^2_{min}=3.84$ for (2) , $conf_{min}=0.7$ for (3) and $2 \leq n_X(r) \leq 8$. When 200 rules for each class was extracted, then the classification was done. $max\{score(k)\} \geq 0.1$ was used for the class label prediction.

Table 1 shows the number of instances gathered into Division B and the classification accuracy. Instances having missing values of A_1-A_{25} and A_{76}-A_{100} were labeled at generation 21 and the number of instances in Division B increased gradually. Eventually, almost the instances having missing values of A_1-A_{50} were labeled. It was found that the prediction accuracy of the instances in Division B was also improved through generations. The accuracy of re-prediction for Division A did not always good compare to Division B. One of the reasons of this can be that the proposed method discovered the more useful information for the classification from the data in Division C. In the case of using instances in JPT and CHB as noise information for Division C, some noise instances were predicted as $C=0$ and the accuracy of Division B ($C=0$) decreased. Using a strict condition like $max\{score(k)\} \geq 0.3$ avoided the wrong labeling, however, the number of labeled instances in Division B ($C=1$) decreased.

[1] http://www.affymetrix.com/support/technical/sample_data/ 500k_hapmap_genotype_data.affx

Table 1: Number of classified instances by evolving classifier.

Generation of GNP	Division (Class)	# classified instances	Accuracy (%)
0 (Round 0)	A (C=1)	30	—
	(C=0)	30	—
21 (Round 1)	A (C=1)	23	95.7
	(C=0)	31	90.3
	B (C=1)	20	90.0
	(C=0)	27	88.9
31 (Round 2)	A (C=1)	23	95.7
	(C=0)	31	90.3
	B (C=1)	55	96.4
	(C=0)	44	95.5
41 (Round 3)	A (C=1)	25	96.0
	(C=0)	30	93.3
	B (C=1)	58	98.3
	(C=0)	49	98.3
113 (Round 10)	A (C=1)	24	91.7
	(C=0)	33	84.9
	B (C=1)	59	100.0
	(C=0)	59	100.0
197 (Round 17)	A (C=1)	26	96.2
	(C=0)	29	93.1
	B (C=1)	59	98.3
	(C=0)	60	98.3

4. CONCLUSIONS

We have proposed an evolving classification method using the GNP based rule extraction. The method evolves the classifier using the labeled instances by itself. The results of experiment using incomplete data showed that the proposed method has a potential of gathering information through GNP generations. The results also showed that the accuracy of the classifier can be improved through generations. We are studying applications of the proposed method to information processing in the traffic systems.

5. REFERENCES

[1] K. Shimada and K. Hirasawa, "A Method of Association Rule Analysis for Incomplete Database Using Genetic Network Programming", in Proc. of the Genetic and Evolutionary Computation Conference 2010 (GECCO2010), pp. 1115–1122, 2010.

[2] K. Shimada, K. Hirasawa and J. Hu, "Class Association Rule Mining with Chi-Squared Test Using Genetic Network Programming", in Proc. of the IEEE Conf. on Systems, Man, and Cybernetics, pp. 5338–5344, 2006.