FTXI: Fault Tolerance XCS in Integer

Hong-Wei Chen
Department of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwam
hweichen@cs.nctu.edu.tw

Ying-Ping Chen
Department of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwam
ypchen@cs.nctu.edu.tw

ABSTRACT

In the realm of data mining, several key issues exists in the traditional classification algorithms, such as low readability, large rule number, and low accuracy with information losing. In this paper, we propose a new classification methodology, called *fault tolerance XCS in integer* (FTXI), by extending XCS to handle conditions in integers and integrating the mechanism of fault tolerance in the context of data mining into the framework of XCS. We also design and generate appropriate artificial data sets for examining and verifying the proposed method. Our experiments indicate that FTXI can provide the least rule number, obtain high prediction accuracy, and offer rule readability, compared to C4.5 and XCS in integer without fault tolerance.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*

General Terms

Algorithms

Keywords

XCS, fault tolerance, integer representation, data mining, classification

1. INTRODUCTION

No matter in what business, there are some history records. For example, a bank has a lot of credit card transactions, and a supermarket has many shopping records. We believe there is certain information in these history records. If we can get such information, we will be able to design better strategy to get higher profit. To achieve this goal, servals solutions were proposed by researchers, and one of those solutions is classification. Classification is a kind of the supervised-learning method and is commonly used by industry. The classic classification algorithms include C4.5, ID3, Artificial Neural Network(ANN), Learning Classifier System(LCS) [1], and XCS [3]. The above algorithms work successfully under their respective hypotheses.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

Although the above algorithms have high accuracy, they still have problems or limitations, such as that some data patterns can not be recognized and that the number of generated rules is too large. In order to handle this situation, we propose a strategy named *Fault Tolerance XCS in Integer* (FTXI), which can provide the least rule number, get high accuracy, and achieve high readability. Particularly, we extend XCS into the integer domain and modify XCS in integer to permit fault tolerance.

In the remainder of this paper, we will discuss in detail about the problems of traditional classification algorithms and how to handle those problems with the mechanism of fault tolerance in section 2. Then, the proposed modification to integrate the mechanism of fault tolerance into XCS will be described in detail in section 3, followed by displaying our experimental results in section 4. Finally, we conclude the present work in section 5.

2. FAULT TOLERANCE IN DATA MINING

In this section, we will first discuss problems of traditional algorithms. Then, we will introduce the idea of fault tolerance in the context of data mining.

2.1 Problem Description

Learning Classifier Systems (LCS) and XCS ignores some noise information to get better accuracy. Take Table 1 as an example, the four rules will be discarded if they are rarely triggered. As a consequence, we lose these rules and the information embedded in them.

2.2 Fault Tolerance

In 2001, Han et al [2] indicated that the real world tends to be diverse and dirty. There are few non-trivial rules with high support and confidence in real data sets. *Support* is the times a rule is triggered. *Confidence* stands for accuracy of a rule. For this reason, we know that knowledge discovery from large real-world data sets needs the mechanism of fault tolerance, as demonstrated in the following examples.

Example 1 : To study students' performance in several courses, we might find the following rules:

R1: good(x,data structure) and good(x,algorithm) and good(x,AI) and good(x,DBMS) → good(x,data mining), where good(x,y): student x get A in course y.

R2: A student who is good in at least three out of four courses: data structure, algorithm, AI, and DBMS is also good at data mining.

Although R1 has a high predict accuracy (high confidence), it cover only a small set of cases (low support). In-

Attribute	DS	ALGO	AI	DBMS	Data Mining
rule1	1	2	3	2	2
rule2	2	2	3	2	2

Table 1: Example of XCS in integer

Attribute	A	B	C	D	FTN	Class level
rule1	1	2	3	2	1	2
rule2	2	2	3	2	0	2

Table 2: Example of FTXI

stead, R2 is more general than R1 because every student satisfies R1 also satisfies R2, but not vice versa. Therefore, R2 gets more support than R1 does. Han et al named that the rule requires data to match only part of its left side as *fault tolerance*. They expect that fault tolerance operation will generate rules with high accuracy and support.

3. FTXI

Traditional XCS takes {0,1,#} as its input, which cannot accommodate the numeric classification problems. Hence, we will introduce a method that can transfer XCS into the integer domain. Then, we discuss how to modify XCS to permit fault tolerance. We call the extended version of XCS as the *fault tolerance XCS in integer*, FTXI.

3.1 XCS in Integer

Although XCSI [4] was proposed in 2001, it does not fulfill our need. We design a simple integer XCS for our own purpose. Using the same idea of XCS, for each attribute we use a numeric value standing for its real meaning. Take Tables 1 as an example, rule1 in Table 1 means if a student gets a good grade in data structure, an average grade in algorithm, a bad grade in AI, and an average grade in DBMS, we can know that his grade of data mining will be average. With this transformation, we can take integers as the input to our system and handle numeric classification problems.

3.2 Fault Tolerance Representation

In the previous section, we tell about how to transfer XCS into the integer domain. Now we will discuss on how to combine XCS with fault tolerance. First, we focus on the representation. We add an attribute, FTN, which records the number of attribute that data do not have to match the rule. Take Table 2 as an example, rule1 means {(A=1) and (B=2) and (C=3) and (D=2)} → (Class level=2) with one error matching attribute still match the fault tolerance rule. In the traditional methods, we must use many rules to represent these patterns, but now we can use one rule to represent these rules and use the sum of these rules fitness as the fault tolerance rule's fitness.

After transferring XCS into integer, applying the fault tolerance operation, the *Fault Tolerance XCS in Integer*, FTXI, is ready for action. We expect FTXI to provide the least rule number and offer high accuracy and readability.

4. EXPERIMENTAL RESULTS

The data set is divided into the training set and the test set. We use the training set to build the prediction model and the test set to test the model accuracy. Moreover, the data set is also tested with C4.5, Integer XCS, and FTXI in ten independent runs.

	XCS in Integer	FTXI	C4.5
Rule Number	35.5	30.6	961
Recognition Rate	99.22%	100%	99.56%
No Recognition Rate	34.06%	0%	0%
Accuracy	65.43%	100%	99.56%

Table 3: Results for fault tolerance sample in the training set

	XCS in Integer	FTXI	C4.5
Recognition rate	99.93%	100%	100%
No recognition rate	34.74%	0%	0%
Accuracy	65.22%	100%	100%

Table 4: Results for fault tolerance sample in the test set

4.1 Results and discussion

The experimental results are shown in Tables 3 and 4. In the beginning, we focus on the rule numbers of FTXI and C4.5. Reviewing Tables 2 and 1, C4.5 generates rules like those in Table 1, and a large rule number is inevitable. However, FTXI uses rules like the first row of Table 2 to represent other potential rules. That is the reason why FTXI uses only 30.6 rules, but C4.5 needs 961 rules.

5. CONCLUSIONS

FTXI can provide the least rule number and get high accuracy and readability. But there still are some problems in FTXI, such as error fault tolerance data which lower the recognition rate, some rules with the same meaning that cannot subsume each other, and automation on fault tolerance number. How to cluster the rule set into smaller set is still an interesting work. How to apply FTXI to association rule mining is a good question waiting to be answered.

Research along this line should be continuously pursued and conducted in order to develop not only feasible but also practical classification systems to further advance the business, science, and even art in the world.

6. ACKNOWLEDGMENTS

The work was partially sponsored by the National Science Council of Taiwan under grant NSC-94-2213-E-009-120. The authors are also grateful to the National Center for High-performance Computing for computer time and facilities.

7. REFERENCES

[1] J. H. Holland. Adaptation. In *Progress in Throretical Biology*, volume 4, 1976.

[2] J. Pei, A. K. H. Tung, and J. Han. Fault-Tolerant Frequent Pattern Mining: Problems and Challenges. In *ACM-SIGMOD*, 2001.

[3] S. W. Wilson. Classifier fitness based on accuracy. *Evolutionary Computation*, 2(2):149–175, 1995.

[4] S. W. Wilson. Mining Oblique Data with XCS. In *Advances in Learning Classifier Systems*, 2001.