규칙기반시스템의 구축에 필요한 규칙 발생 기법

정보위*, 여정모**
*부경대학교 정보공학과
**부경대학교 컴퓨터공학과
e-mail : pknu2008@pknu.ac.kr; yeo@pknu.ac.kr

The method of making Rule Cases to build Rule-Based System

BaoWei Zheng*, Jeongmo Yeo**
* Dept. of Information Engineering, Pukyong National University
**Dept. of Computer Engineering, Pukyong National University

요 약

트리 유형의 규칙들을 처리하는 기존의 규칙기반시스템은 실제의 규칙들을 절차형 프로그램으로 구성된 규칙 엔진에게 제공하여 결과값을 반환받는 형식으로 동작한다. 이와 같은 방식은 두 가지 단점이 있는데, 그 하나는 업무의 변경에 따라 규칙 엔진을 변경해야 한다는 점이고, 또 하나는 엄청나게 많은 규칙들을 가진 경우에는 규칙 엔진이 복잡해지고 규칙 엔진의 속도가 저하된다는 점이다. 본 연구에서는 ID 트리의 원리를 적용하여 규칙기반시스템에 사용되는 규칙들을 생성하는 규칙간소화 알고리듬을 제안한다. 제안하는 알고리듬은 규칙기반시스템에 필요한 최소의 규칙들을 생성할 수 있을 뿐 아니라 업무가 변경되는 경우 알고리듬의 수행으로 쉽게 규칙들을 생성할 수 있으므로 업무변화에 유연하다. 그리고 규칙 엔진이 필요하지 않아 수행속도의 향상과 경비 절감의 효과도 기대한다.

Abstract

Tree type of Rule Case will be processed by the method that provide practical Rule Case to Rule Engine that is made with procedural language beforehand, then the Rule Engine according to the condition of the special Rule Case to return result in current Rule-Based System. There are two disadvantages in the method; the first is according to specific business rule after construct the Rule Engine when the business rule changing the Rule Engine also must be changed. The second is when Rule have many conditions the Rule Engine will become very complex and the speed of processing Rule Case will become very slow. In this paper, we will propose a simplified algorithm that according to the theory of ID Tree to produce Rules which be used in Rule-Based System. The algorithm can not only produce Rules but also make sure of satisfying change of business rule by execute the algorithm. Because it is not necessary to make a Rule Engine, we will anticipate effect of increasing speed and reducing cost from Rule-Based System of applying the algorithm.

1. Introduction

Rule-Based System is defined as represent knowledge in terms of a bunch of rules that tell us what we should do or what we could conclude in different situations. A general Rule-Based System consists of a bunch of rules, a bunch of facts, and some interpreter controlling the application of the rules, given the facts. There are three types of Rule in the Rule-Based System, such as multi dimension, rule matrix and rule tree. We have researched out a simplified algorithm that can simplify amount of rule case for the type of multi dimension and rule matrix in previous paper. In this paper, we will propose another new algorithm that according to the theory of ID Tree to build Rule-Based System and simplify rule cases.

An identification tree is a representation. That is a decision tree in which each set of possible conclusions is established implicitly by a list of sample data of known class. The smallest identification tree that is consistent with the sample is the one that is most likely to identify unknown rule Case correctly and return a result of practical Rule Case. Identification Tree building is the most widely used learning method, so we make use of the characteristics of it to building Rule-Based System.

Because disadvantages of when the business rule changing the Rule Engine also must be changed and Rule Engine is very complex, they make the Rule-Based System cannot be applied widely in many area. However, we propose a new algorithm that can solve these problems and extend applied

area of Rule-Based System.

2. Related work

When results of some functions or conditions are preconditions of other functions or conditions, and make use of these characteristics can building a tree. Every condition is considered as a node of tree and include more than function into a node. In every node through compare conditions or function calculus return a temporary result, then other node execute itself result by reference the below node returned result. Propositional logic calculus operation must be executed by special mechanism in Tree type of Rule-Based System. The special mechanism that is made by procedural language can complete all of tasks and it is called Rule Engine. In order to process Tree type of rule we must develop a rule engine for completing propositional logic calculus operation. However, the biggest disadvantage of Rule Engine is what it is not changed easily following the change of Business Rule. Because Business Rules are always updated, in order to let the Rule-Based System adapt to these change quickly we research a method that have not need to develop Rule Engine process Tree Type of rule. Architecture of processing the Tree Type of rule in current Rule-Based System as followed Fig.1.

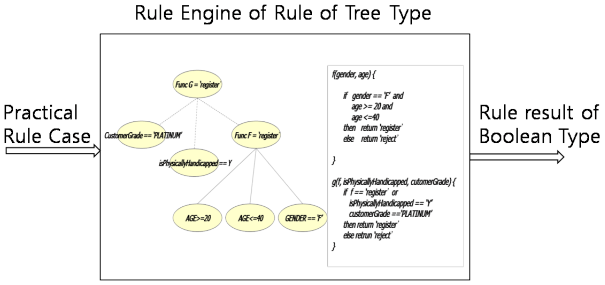

Fig.1. Architecture of processing rule of tree type

Inputting a practical Rule case into Rule Engine and return a Rule result of Boolean type, the Rule Engine is only a program that is made by procedural language.

According to the characteristic of Rule of Tree type data model can be designed as followed Fig.2.

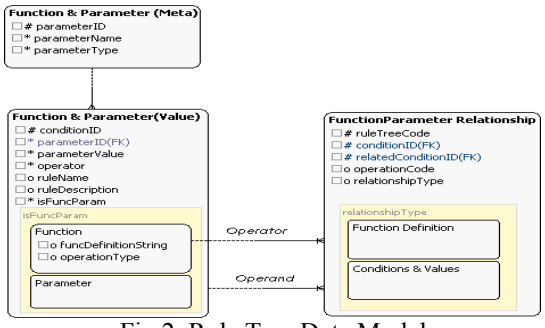

Fig.2. Rule Tree Data Model

There are three entities type in the Rule Tree Data Model. Although the data model represent all of function and parameter, values of function and parameter and relationship of them, but these information cannot help us to increase speed of executing and instead of Rule Engine.

3. Building RBS with Identification Tree

Identification Tree (ID Tree) is a decision tree in which all possible divisions is created by training the tree against a list of known data. The purpose of an ID Tree is to take a set of sample data, classify the data an construct a series of test to classify an unknown object based on like properties.

First the tree must be created and trained. 1) It must be provided with sufficient labeled samples that are used to create the tree itself. 2) It does this by dividing the samples into subsets based on features. The sets of samples at the leaves of the tree define a classification. The tree is created based on Occam's razor, which states that the simplest tree, that is consistent with the training samples, is the best predictor. To find the smallest tree, one could find every possible tree given the data set then example each one and choose the smallest. However, this is expensive and wasteful. Therefore, the solution to this is to greedily create one small tree. The process of training a tree as following:

1) At each node, pick a test such that branches are close to same classification.
2) Split into subset with the least disorder.
3) Find which of these tests minimizes the disorder.

Then until each of leaf node contains a set that is homogenous or is near homogenous. Select a leaf node that is non-homogenous split this set into two or more homogenous subsets to minimize disorder. Since the goal of an ID Tree is to generate homogenous subsets, we want to calculate how non-homogenous the subsets each test creates. The test that minimizes the disorder is the one that divides the samples into the cleanest categories. Disorder is calculating as follows:

Average disorder=Σ_b (nb/nt) * (Σ_c (nbc/nb)$\log_2$(nbc/nb))

Where: nb is the number of samples in branch 'b', nt is the total number of samples in all branches, nbc is the total of samples in branch b of class c.

For a real database of any size, it is unlikely that any test would produce even one completely homogenous subset. Accordingly, for real database, one needs a powerful way to measure the total disorder, or in homogeneity, in subsets produced by each test. Information theory can be used to compute a measure disorder.

To see why this borrowed formula works, one needs to focus on the set of samples lying at the end of one branch b. what is required here is a formula involving nb and nbc that gives a high number when a test produces highly inhomogenous sets and a low number when a test produces completely homogenous sets. The following formula involving nb and nbc generall works:

Disorder = c (nbc/nt) log2 (nbc/nb)

The architecture of making Rule Case to build Rule-Based System as followed Fig.3:

In this architecture we save all rules into database table, when user through SQL or application submit a practical Rule Case to database, the database according to the attributives of the Rule Case search a corresponding result data from table. This technology of searching Rule Case result like is similar to search a general data of satisfying many condition data from database. Because we used technology is database technology, all of scanning technology, all of index technology, and clustering technology and so on can be used for return result of Rule

Case. Therefore, speed of return result of Rule Case will be increased to a great extent.

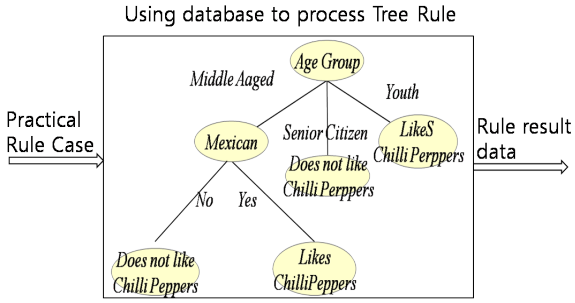

Fig.3. Architecture of making Rule Case to build RBS

The following are the observations of an astute gastronome about the reaction of group diners of different ages, weight and height, enjoying chili peppers in Korea restaurant, Koreans obviously enjoy their nation dish, but non-Koreans appear to have difficulty with the dish:

Table.1 example for surveying Korean like chilli peppers

Name	AgeGroup	Height	Weight	Korean	Like?
Diana	M-Aged	Average	Light	No	No
Jose	M-Aged	Short	Tall	Yes	Yes
Zapata	Youth	Tall	Average	Yes	Yes
Charles	M-aged	Short	Average	No	No
Philip	S-Citizen	Average	Heavy	No	No
Zara	Youth	Tall	Heavy	No	Yes
April	Youth	Average	Heavy	No	Yes
Pablo	M-Aged	Short	Light	Yes	Yes

An Identification tree based on age group, height, weight and nationality (whether or not the individual is Korean) on the above chilli pepper database is shown below:

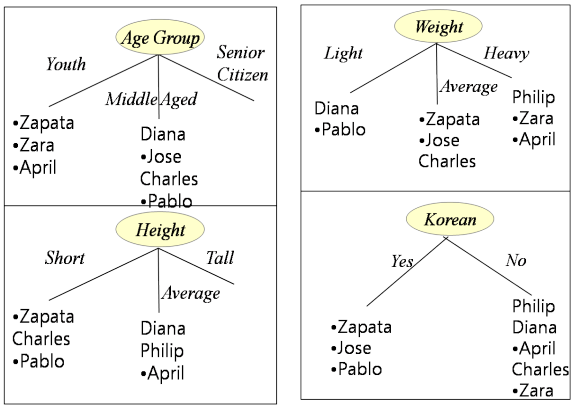

Fig.4. Training ID Tree for every attributive

If the question now is this: what happens when you only select middle aged people from the database? Once the middle-aged people are isolated the available tests performances.

In Fig.4 the back point in front of people name represent Age group is equal middle aged. Because when we select people Age group='Middle-aged' as a condition, keep all of data that satisfy the condition in every tree except Age Group tree. We delete every data that doesn't satisfy the condition from the tree. Then we accept a simplified ID tree such as showing in Fig.5

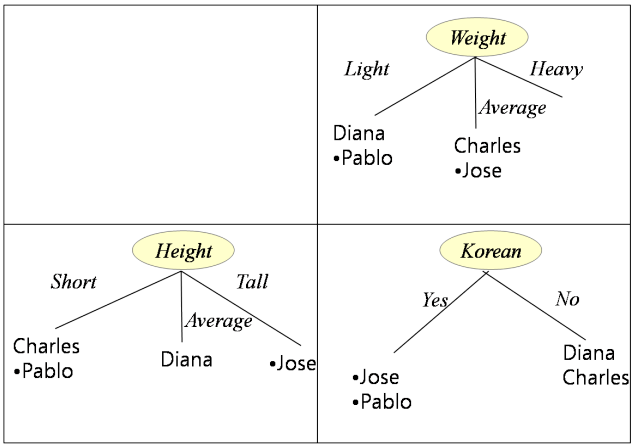

Fig.5. ID Tree of satisfying specific condition

4. Simplified algorithm

Now if we compute the average disorder produced by the age-group test and on the basic of person's Korean origin. We will find that age and nationality test ensure proper identification for all the samples shown in the table.

Average Disorder (Height) = 3/8 (-1/3 log2 (1/3) -2/3 log2 (2/3))3/8 (-2/3 log2 (2/3) -1/3 log2 (1/3))=0.69

Average Disorder (Weight) = 2/8 (-1/2 log2 (1/2) -1/2log2 (1/2))3/8 (-1/3 log2 (1/3) -2/3 log2 (2/3))3/8 (-2/3 log2 (2/3) -1/3 log2 (1/3))=0.94

Average Disorder (Nationality) = 5/8 (-3/5 log2 (3/5) -2/5log2 (2/5))3/8 (-0 log2 (0) -3/3 log2 (3/3))=0.61

Average Disorder (Age Group) = 4/8 (-2/4 log2 (2/4) -2/4 log2 (2/4))1/8 (-1/1 log2 (1/1) -0 log2 (0))3/8 (-3/3 log2 (3/3) -0 log2 (0))=0.5

The age group and nationality test are by far the best discriminators in that their use together will ensure the proper identification of all the samples. Now if we focus exclusively on the middle aged people we get the following results:

Table.2 Disorder of exclusive the middle age

Test	Disorder1	Disorder2
Nationality	0.61	0
Height	0.69	0.5
Weight	0.94	1

We can make a procedure for generating identification trees, this procedure helps in the generation of an identification based on the computation of disorder introduced by each of the concepts.

Until each leaf node is populated by as homogeneous a sample set as possible disorder age group and Korean. Then select a leaf node with an inhomogeneous sample set, replace that leaf node by a test node that divides the inhomogeneous sample set into minimally inhomogeneous subsets, according to some measure of disorder.

Then make a procedure pruner for converting an identification tree into rule sets. This procedure helps in the

conversion of an identification tree into a rule set: create one rule for each root-to-leaf path in the identification tree, and then simplify each rule by discarding antecedents that have no effect on the conclusion reached by the rule. Replace those rules that share the most common consequent by a default rule that is triggered when no other rule is triggered. In the event of a tie, use some heuristic tie-breaker.

Now if we are asked to construct an identification tree for determining peoples appetite for chilli peppers, we will come up with the simplest ID Tree which represents the data is shown as Fig.5.

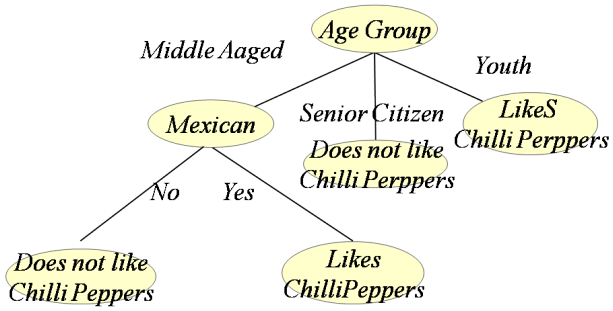

Fig.5. The simplest Identification Tree

Finally, the following rule set can be derived using Pruner:
IF age_group is middle_aged and nationality is not Korean
Then the person does_not_like chilli pepper, if age_group is senior_citizen then the person does_not like chilli pepper, and if no other rule apples then the person likes chilli pepper.

Now we have introduced the whole procedural of the algorithm. The internal architecture of the algorithm can be designed as the Fig.6. The detail process of the simplified algorithm can be shown in the Figure.

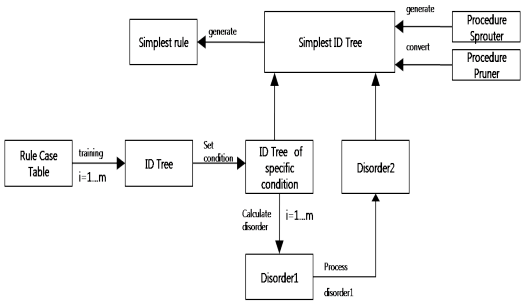

Fig.6. Architecture of simplified algorithm

In the figure, according to every attributives of the Rule Case Table train a Identification Tree. Then set a special condition for every Identification Tree and delete every data that does not satisfy the condition from Identification Tree. Calculating disorder for every attributive with above formula, and compare each disorder. Through compare and analysis the disorder that calculated in previous step then set them the disorder as disorder2. Using the Disoder2 and run procedural sprouter and pruner for simplifying the ID Tree to obtain a Simplest Tree. Final if it is need it can generate all of simplest rules from the simplest ID Tree.

5. Conclusion

In this paper we introduced a new architecture of Rule-Based System and one types of Rule Data model. It is very advantageous to make use of elements of Database technology for simplifying the operation of Rule-Based System. The combination of Rule-Based System elements and Relational Database technology can produce potential practical significance in to practical application of Rule-Based System. According to the characteristics of rule we have designed three kinds of Rule Data Model, they can be applied into any situations where we wish to capture data of rules. In order to advance the efficiency of processing large rule case we proposed a Simplification Algorithm, and proved the correctness and reasonableness of all methods of referring to in the algorithm. .

Although we can make use of the Simplification Algorithm to advance the efficiency of processing rule case in a great degree, it can be used in specific situation rather than all situations. Thus we should extend applied range of the Simplification Algorithm to all of possible situation and advance further shortages of the Simplification Algorithm and make it more perfect in the future

Reference

[1] En-core consulting. Rule Base Data Model Seminar, 2008. www.en-core.com.

[2] Editor-in-chief and Prof. Janusz Kacprzyk, “Logical Functions for Rule-Base Systems, second edition”, Spinger, 2006.

[3] Ahmed T. Sadik, “Premises Reduction of Rule Based Expert System Using Association Rules Technique”, International Journal of Soft Computing 3(3): 195-200, 2008.

[4] Oracle Corp. Oracle® Fusion Middleware User's Guide for Oracle Business Rules 11g Release 1 (11.1.1), 2009.

[5] Alison Cawsey, “The architecture of forward chaining Rule-Base System and backward chaining Rule-Base System”, Department of Computing and Electrical Engineering Heriot-Watt University Edinburgh EH14 4AS, UK, 2007.

[6] David C.Hay, “Data Model Patterns A Metadata Map”. Morgan Kaufmann Publishing, 2007.

[7] Len Silverston, Paul Agnew, “The Data Model Resource Book”, Wiley Publishing, Inc. 2009, pp. 411-468.

[8] Steve Hoberman & Associates, “LLC. Data Modeling Master Class. 2008”, pp. 112-280, www.stevehoberman.com.

[9] Stephande Faroult & Peter Robson, “The Art of SQL. Publishing House of electronics industry”, 2008, pp.167-190.

[10] Lee Huw Sick, “New Written, Large Scale Database Solution. Publishing En-core consulting”, 2005, pp.1-99, 323-399.

[11] Malcolm Chisholm, “From How to Build a Business Rules Engine: Extending Application Functionality through Metadata Engineering”, Morgan Kaufman, 2004.

[12] L.Silverston, “The Model Resource Book, Revised Edition, Volumen1, A library of Universal Data Models for All Enterprise”, Wiley, pp.133-180, 2001.

[13] Dr.Graeme Simsion, “Data Modeling Theory and Practice” Technics , LLC and, 2007.