Data Mining Methods for Rule Designing and Rule Triggering in Active Database Systems

Nasrin Kalanat, Mohammad Reza Kangavari

The Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
kalanat@comp.iust.ac.ir, kangavari@iust.ac.ir

Abstract

Active database system has been introduced to extend the database functionality. It is capable of detecting specific events and automatically reacting to them by executing certain actions either inside or outside the database. This behavior is usually specified through Event Condition Action (ECA) rules. Rule design plays a critical role in implementing an active database system. It is not always straightforward due to lack of methodology. In this paper a new approach is presented to identify ECA rules that could strongly express application semantics. Additionally an ensemble method is proposed to trigger more appropriate ECA rules when the interesting events occurred.

Keywords: *Active data base system, ECA rule, Associative classification rule, Bagging*

1. Introduction

Traditional database systems (DBSs) have been developed to store and manage large volumes of data. During the last decade, one of the trends in database technology has been focused to extend these database systems to enhance their functionality and to accommodate more advanced applications. Extending database systems with powerful rule-processing capabilities has been one of these enhancements. Rule-processing capabilities can be divided into two classes: deductive, in which logic-programming-style rules are used to provide a more powerful user interface than that provided by most database query languages [1], and active, where production-style rules are used to provide automatic execution of predefined operations in response to the occurrence of certain events [2,3]. The latter supports active behavior of a system that is generally specified by means of rules. These rules are capable of specifying events that have to be detected, conditions that have to be evaluated, and reactions of the system when the events occurred. Overall, the rules consist of an event, a condition and an action part, known as Event-Condition-Action rules or ECA rules. Those Database systems that are enhanced with active capabilities are known as active databases systems (ADBSs). The active behavior of an ADBS drives from application semantics. Thus application semantics are partly expressed in ECA rules and managed by an ADBS [4, 5]. Consequently rule design plays a critical role in implementing an ADBS to determine which real world policies should be represented within the rules. Traditionally, the rules are specified according to the experiences of experts. Rule design is an analysis activity that assumes an understanding of the universe of discourse. It is practically impossible to guarantee a thorough correspondence between traditionally designed rules set and user expectations. In other word it is practically impossible to guarantee the rules could precisely express the application semantics [6]. Consequently finding efficient rules by traditional way can be difficult or even impossible in different areas of applications [7]. Data mining

techniques can be used to automatize the process. To the best knowledge of the authors, there is no approach to automate the ECA rules design. In this paper the associative classification rule mining is used to extract ECA rules of a data set representing system functions. The approach can find the rules more precisely. Therefore the rules better represent application semantics. Additionally an ensemble method is proposed to trigger more appropriate rules when interesting events occurred. The method utilizes bagging approach to construct the ensemble that consists of associative classifiers built again on bootstrap samples. Rest of the paper is as follows: section 2 contains the preliminaries and definitions needed for later sections. Section 3 explains the ECA rule extracting approach. Section 4 describes a bagging method for triggering appropriate ECA rules. Finally section 5 concludes the paper.

2. Problem Definition

The active behavior of a system is generally described through active rules. These rules specify the interesting events, conditions that have to be evaluated, and reactions of the system when the events occurred. Thus an active rule has three parts: event, condition and action. The event-condition-action rule (ECA rule) is one of the most prevalent models of such rules that can be stated as follows [8]: on event e if condition c then action a.
Event: This part describes the situation of interest in which the rule responds.
Condition: This part specifies what has to be checked once the rule is triggered and before it is executed. If the condition is evaluated as true, then the condition is satisfied.
Action: This part describes the task to be carried out by the rule if the relevant event occurs and the relevant condition is satisfied.
Event and condition parts of each ECA rule can be roughly subdivided into two categories: primitive which is corresponded to elementary one, and composite which is composed out of other composite or primitive ones.

The data set is gathered by some experts. Each expert represents system functions as instances of data set. Each instance is a transaction that contains [attribute, integer-value] pairs and a class label. Each [attribute, integer-value] pair is called an item. Each item in turn might be an event or a condition. Each action is assumed as a class label. An attribute can be a categorical (or discrete) or a continuous (or numeric) attribute. In the presented paper, all the attributes are treated uniformly. For a categorical attribute, all the possible values are mapped to a set of consecutive positive integers. For a continuous attribute, its value range is discretized into intervals, and the intervals are in turn mapped to consecutive positive integers.

3. ECA Rule Extracting Approach

Associative classification rule mining is used for extracting ECA rules that more precisely represent application semantics. This approach is used to determine those events and conditions that the system should pay attention to them. It discovers interesting and critical relations between event-condition and action parts. It intends to identify strong ECA rules discovered in dataset using different measures of interestingness.

3.1. Associative Classification Rules

Utilizing the association rules for classification has been previously described in literature [9]. In associative classification, the focus is to produce association rules

that have only a particular attribute in the consequent. These produced association rules are called associative classification rules (ACRs).

Associative classification differs from general association rule mining by introducing a constraint as to the attribute that must appear on the consequent of the rule. The produced rules can be used to build a model or classifier. ACRs are a particular case of constrained association rules. There have been researches in this area about integrating (pushing) these constraints into the mining phase rather than filtering the enormous number of rules produced using the constraints as post-processing filters. The paper [10] on this area proposes different ways of pushing the constraints into the mining phase. The general advantages are faster execution and lower memory utilization.

In this paper CBA-RG algorithm [9] is used that is an extension of the Apriori algorithm. The goal of this algorithm is to find all rule items of the form $< condset;$ $y >$ where $condset$ is a set of items, and $y \in Y$ where Y is the set of class labels. The support of the rule item is the number of instances in the data set that contain the $condset$ and are labeled with y. Each rule item is corresponded to a rule of the form: $condset \rightarrow y$.

Rule items that have support equal or greater than to $minsup$ (a user defined threshold) are called frequent rule items, while the others are called infrequent rule items. For all rule items that have the same $condset$, the one with the highest confidence is selected as the representative of those rule items. The confidence is, out of all the instances in the data set that contain $condset$, the percentage that are labeled with y as well. Confidence of a rule can be computed as support count of the rule item $\div$ support count of the $condset$. The confidence of a rule measures the strength of the rule (correlation between the antecedent and the consequent) while the support measures the frequency of the antecedent and the consequent together. The confidence of a rule item is calculated to determine if the rule item meets $minconf$ (a user defined threshold). The set of rules that are selected after checking for support and confidence is called the ACRs.

4. Bagging Approach for Rule Triggering

Proposed approach is an ensemble method that triggers more appropriate ECA rules when the interesting events occurred. The method uses the bagging approach for constructing the ensemble that consists of associative classifiers built again on bootstrap samples. The method combines ensemble votes for triggering more appropriate ECA rules when the interesting events occurred.

4.1. Bagging Rule Triggering Method

Bagging is the generation of several models from bootstrap samples of the same original dataset [11]. The prediction given by the set of resulting models for one example is done by majority vote strategy or by averaging the predictions of the different models. Bagging has the effect of improving the results of an unstable classifier by reducing its variance [12].

In the case of associative classification, produced rules can be used to build a model or classifier. So proposed approach takes bootstrap samples of the data set and trains an associative classifier on each of them. Finally an ensemble of models is obtained. The models differences tend to reflect the variability of ECA rule sets obtained from the same source of data.

When interesting events occurred, each model of the ensemble nominates the ECA rule/s to be triggered. The method combines candidate rules of ensemble models by obtaining frequent sets of condition-action parts of the rules. An Apriory-

like method is used for this purpose. Finally more appropriate rule/s will be triggered.

Now the BRT (Bagging Rule Triggering) algorithm (Algorithm 1) is described in detail.

Training phase

1. Initialize the parameters
   - D, the data set that is set of instances of system functions gathered by experts.
   - $MSup$: minimum value for support parameter.
   - L, the number of classifiers to train.
   - $E=\emptyset$, the ensemble, $R=\emptyset$, the rules set.
2. For $k = 1, \ldots, L$
   - Take a bootstrap sample S_k from D.
   - Build an Associative classifier E_k (set R_k contains ACRs) using S_k as the training set.
   - Add the classifier to the current ensemble, $E = E \cup E_k$ ($R = R \cup R_k$):
3. Return E, R.

Triggering phase

4. Run $E_1, \ldots, E_L$ When the interesting events occurred and generate candidate rules set CR for triggering.
5. Find condition-actions parts of CRs that have support greater than or equal to $MSup$ then trigger their rules.

Algorithm 1. BRT (Bagging Rule Triggering) algorithm.

5. Conclusion

The implementation of an active application requires many complex rules to specify the system's active behavior. Rule design is very important for an active database system, and it is also difficult for us. In order to cope with problems arisen in active rule design, in this paper an associative classification method was originally introduced into the process of active rule design. Also a bagging strategy was introduced for rule triggering. It produces an ensemble of associative classifier for triggering rules precisely.

References

[1] S. Ceri, G. Gottlob and L. Tanca, "Logic programming and databases", Springer, **(1990)**.
[2] Act-Net Consortium, "The active database management system manifesto: A rulebase of ADBMS features", ACM SIGMOD Record, vol. 25, **(1996)**, pp. 40-49.
[3] U. Dayal, A. Buchmann and D. McCarthy, "Rules are objects too, In Lecture notes in computer science", Proceedings of the 2nd International Workshop on Object-Oriented Database Systems, vol. 334, **(1988)**, pp. 129-143.
[4] A. Vadua, "Rule Development for active database", PhD. Thesis, CS Department, University of Zurich, **(1999)**.
[5] D. Goldin, S. Srinivasa and V. Srikanti, "Active databases as information system", Database Engineering and Applications Symposium, **(2004)**, pp. 123-130.
[6] M. Dai and Y. Huang, "Data Mining Used in Rule Design for Active Database Systems", Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007, **(2007)**, pp. 588-592.
[7] R. Trepos, A. Salleb-Aouissi, M. Cordier, V. Masson and Ch. Gascuel-Odoux, "Building actions from classification rules", Knowledge and Information Systems Journal, vol. 34, **(2013)**, pp. 267–298.
[8] A. Silberschatz, H. F. Korth and S. Sudarshan, "Database System Concepts", 3rd Ed. New York: McGraw-Hill, **(1999)**.
[9] B. Liu, W. Hsu and Y. Ma, "Integrating classification and association rule mining", Knowledge Discovery and Data Mining Integrating, **(1998)**, pp. 80-86.

[10] R. Srikant, Q. Vu, and R. Agrawal, "Mining association rules with item constraints", D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors, Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining, KDD, AAAI Press, vol. 14, **(1997)**, pp. 67-73.

[11] L. Breiman, "Bagging Predictors", Machine Learning, vol. 24, **(1996)**, pp. 123-140.

[12] T. Hastie, R. Tibshirani and J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference and Prediction", Series in Statistics, Springer, **(2001)**.

[13] M. J. Alıpio and P. J. Azevedo, "An Experiment with Association Rules and Classification: Post-Bagging and Conviction", LNAI, 3734, Springer, **(2005)**, pp. 137–149.