CBC: An Associative Classifier with a Small Number of Rules[☆]

Houtao Deng

Intuit, Mountain View, CA, USA

George Runger

Arizona State University, Tempe, AZ, USA

Eugene Tuv

Intel, Chandler, AZ, USA

Wade Bannister

Ingenix Consulting, Irvine, CA, USA

Abstract

Associative classifiers have been proposed to achieve an accurate model with each individual rule being interpretable. However, existing associative classifiers often consist of a large number of rules and, thus, can be difficult to interpret. We show that associative classifiers consisting of an ordered rule set can be represented as a tree model. From this view, it is clear that these classifiers are restricted in that at least one child node of a non-leaf node is never split. We propose a new tree model, i.e., condition-based tree (CBT), to relax the restriction. Furthermore, we also propose an algorithm to transform a CBT to an ordered rule set with concise rule conditions. This ordered rule set is referred to as a condition-based classifier (CBC). Thus, the interpretability of an associative classifier is maintained, but more expressive models are possible. The rule transformation algorithm can be also applied to regular binary decision trees to extract an ordered set of rules with simple

[☆] This research was partially supported by ONR grant N00014-09-1-0656.

Email addresses: hdeng3@asu.com (Houtao Deng), george.runger@asu.edu (George Runger), eugene.tuv@intel.com (Eugene Tuv), wade.bannister@ingenixconsulting.com (Wade Bannister)

rule conditions. Feature selection is applied to a binary representation of conditions to simplify/improve the models further. Experimental studies show that CBC has competitive accuracy performance, and has a significantly smaller number of rules (median of 10 rules per data set) than well-known associative classifiers such as CBA (median of 47) and GARC (median of 21). CBC with feature selection has even a smaller number of rules.

Keywords: association rule; decision tree; feature selection; rule-based classifier; rule pruning.

1. Introduction

The comprehensibility of a classifier is vital in decision support systems [18, 12]. An interpretable classifier can provide insights about why objects are classified into a specific class. For example, [27] employed classification rules to understand the patterns causing patients' hospitalization. By using the explicit knowledge from a comprehensible classifier, decision makers can take actions to improve the existing system [21]. Furthermore, domain experts and decision makers are also able to validate if a comprehensible classifier is applicable in practice. This is particularly useful when the data are messy, the amount of data is not large enough, or the decision makers are not comfortable in completely relying on data-driven models.

Decision tree classifiers such as C4.5 [20] and CART [4] are easy to understand, and have been popularly used. However, decision trees usually consider only one variable at each step/node, and tend to be greedy and have difficulty in capturing strong variable interactions, such as an exclusive OR (XOR) relationship where an individual attribute is not predictive, but the combination can be effective.

An associative classifier [17] is another type of comprehensible and popularly used classifier. The associative classification rules are generated by association rule algorithms, such as the Apriori algorithm [1] , but with the right hand side of the rule being fixed as the class attribute. An associative classification rule is just a IF-THEN rule. For example, $\{(outlook = sunny \wedge wind = false) \Rightarrow class = play\ tennis\}$ implies the target class is *play tennis* if the attributes *outlook* is *sunny* and *wind* is $false$. Because associative classification rule-generating algorithms usually consider multiple variables simultaneously, a classifier formed by these rules can be less greedy than decision trees. Indeed, previous work has successfully shown the competitive accuracy performance of associative classifiers, such as CMAR [16], CPAR [28], HARMONY [24], LAC [23] and GARC [5]. Associative classification has been popularly applied in practice, such as detecting products missing from the shelf [19], maximizing customer satisfaction [14] and predicting patients' hospitalization [27].

However, in the associative classification rule generating process, a large number of rules can be generated, and, thus, there can be many redundant or irrelevant rules. Pessimistic error estimation and χ^2 testing were used for pruning rules, but these methods can still leave a large number of rules unpruned [17, 16]. Berrado and Runger [2] used meta-rules to organize and summarize rules. Zaïane and Antonie [29] proposed a visual and interactive user application for pruning rules, which can be useful when domain knowledge is present. Wang et al. [25] pruned rules using a decision tree structure, and Chen et al. [5] pruned rules using information gain and certain conflicts/redundancy resolution strategies. These methods reduced the number of rules in classification *to some degree*. An associative classifier with a small number of rules is often favorable over another with a larger number of rules with similar accuracy performance [5].

We show that associative classifiers consisting of an ordered rule set can be represented as a tree model. This tree tests a rule condition at a node, instead of testing a single variable (ordinary trees). However, in the tree at least one child node of a non-leaf node is never split further. This may limit the expressiveness of the tree. Motivated by the limitation, we propose a condition-based tree (CBT) that relaxes the restriction on splits. Furthermore, we show that CBT can be *equivalently* transformed to an ordered rule set with simplified conditions, referred to as a condition-based classifier (CBC). Thus, the interpretability of an associative classifier is maintained, but more expressive models are possible. We also provide procedures of constructing CBT and CBC. Experimental studies show that CBC has significantly fewer rules than existing associative classifiers CBA and GARC, with similar accuracy performance. Also, feature selection is applied to simplify/improve the models further.

The remainder of this paper is organized as follows. Section 2 presents ordinary decision trees and associative classifiers. Section 3 introduces CBT and CBC. Section 4 discusses the experimental results and Section 5 draws the conclusions.

2. Decision Trees and Associative Classifiers

2.1. Definitions

Let $X = (X_1, ..., X_T)$ be the predictor variables, Y be the class variable, and Y_k $(k = 0, ..., K-1)$ be the class labels. Let $D = \{(x_i, y_i)|i = 1...N\}$ denote a training data set, where x_i and y_i are realizations of X and Y. A classification rule r_j describing D can be expressed as $\{c_j \Rightarrow Y = Y_k\}$, and c_j, referred to as the condition of r_j, is a conjunction of attribute-value pairs, e.g., $(X_1 = \mathit{small} \wedge X_2 = \mathit{green})$. The support of the rule $\{c_j \Rightarrow Y = Y_k\}$ is the proportion of rows in D that satisfy both c_j and $Y = Y_k$. The support of the condition c_j is the proportion of rows in D that satisfy c_j. The confidence of the rule is the ratio between the rule support and the rule condition support.

2.2. Classification Rules from Decision Trees

We consider decision trees that split a non-leaf node into M $(M \geq 2)$ child nodes according to attribute X^* by maximizing the information gain of Y, where $M = 2$ if X^* is continuous, and $M = |X^*|$ if X^* is categorical. Let $p(y_k|P_v)$ denote the proportion of instances belonging to class y_k at node v. The entropy of Y at a node v can be expressed in terms of prior probabilities as

$$H(Y|P_v) = -\sum_k p(y_k|P_v) \log_2 p(y_k|P_v)$$

where P_v is the conjunction of the attribute-value pairs along the path from the root node to v, e.g., $P_v = \{X_1 = \mathit{large} \wedge X_2 = \mathit{green} \wedge X_5 > 10\}$. If v is a non-leaf node, the split variable X^* is selected by maximizing the information gain

$$\mathit{Gain}(Y|P_v, X^*) = H(Y|P_v) - \sum_m w_m H(Y|P_v, X^* = a_m)$$

where w_m is the ratio of the number of instances at child node v_m to the total number of instances at v, and the a_m's denote the values of the categorical attribute X^* (with an appropriate revision for a numerical attribute). If v is a leaf node, it is assigned with the class Y_k that is most frequent in the node, and a classification rule can be formed as $P_v \Rightarrow Y_k$. The rules from a decision tree are mutually exclusive and can be arranged in any order. Figure 1 illustrates a decision tree and the rules for the decision tree are as follows

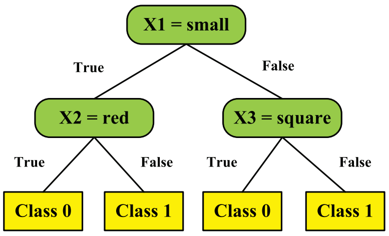

Figure 1: A regular binary decision tree.

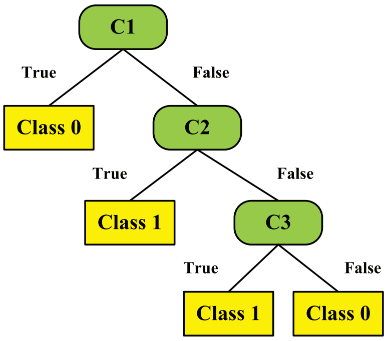

Figure 2: A tree presentation of a set of ordered associative classification rules.

$$\begin{aligned}
&If\ X_1 = small \ \wedge\ X_2 = red &&\Rightarrow\ Y = class\ 0;\\
&If\ X_1 = small \ \wedge\ X_2 \neq red &&\Rightarrow\ Y = class\ 1;\\
&If\ X_1 \neq small \ \wedge\ X_3 = square &&\Rightarrow\ Y = class\ 0;\\
&If\ X_1 \neq small \ \wedge\ X_3 \neq square &&\Rightarrow\ Y = class\ 1.
\end{aligned}$$

2.3. Associative Classifier

In associative classification, classification rules satisfying a certain support and confidence value are generated by an algorithm such as Apriori [1]. Associative classifiers such as CBA [17] are formed by ordering the rules according to some criteria such as the rule confidence and support. For example,

$$\begin{aligned}
c_1 &\Rightarrow Y = class\ 0;\\
c_2 &\Rightarrow Y = class\ 1;\\
c_3 &\Rightarrow Y = class\ 0;\\
Else &\Rightarrow Y = class\ 1.
\end{aligned}$$

where c_j is a rule condition (a conjunction of attribute-value pairs). If an instance satisfies the conditions of multiple rules, the first rule is used. Therefore, the order of the rules affects the classification results. The ordered rules can be presented in a form of tree shown in Figure 2. It can be seen that the left child nodes are not split further.

3. Condition-based Tree and Classifier

Decision trees do not necessarily produce rules with the maximum confidence as they only seek locally optimal solutions, while associative classifiers

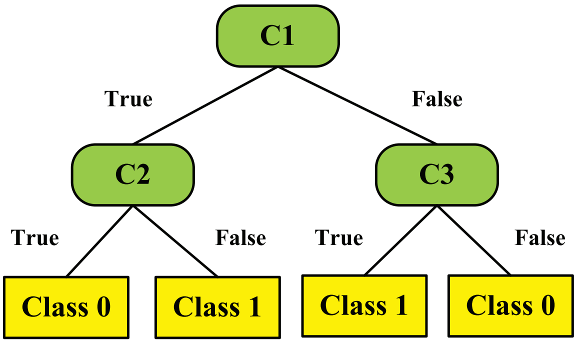

Figure 3: A condition-based tree where c_j is a conjunction of attribute-value pairs, e.g., $X_1 = \mathit{small} \wedge X_2 = \mathit{red}$.

can include all the rules above a minimum confidence. Decision trees select one variable at a time and, therefore, are challenged by some patterns. For example, a challenge occurs when all attributes are needed for a rule (e.g., the XOR logic). However, associative classification considers multiple attributes simultaneously and, therefore, can find the combination of several attributes useful. However, as seen in Figure 2, for associative classifiers consisting of an ordered rule set (e.g. CBA), the child nodes satisfying the splitting criterion are always leaf nodes. This may limit the expressiveness of the tree.

Motivated by the restriction, we propose a new tree model, referred to as a **condition-based tree (CBT)**, illustrated in Figure 3. In a non-leaf node at a CBT, data are split by testing if a condition is satisfied, where a condition is a conjunction of attribute-value pairs. A CBT relaxes the restriction of CBA that at least one of the child node is never split, and, thus, can be more expressive. Also, a CBT relaxes the restriction of ordinary decision trees that the condition tested at each node is limited to one attribute.

Furthermore, in the following sections we show the CBT can be transformed to a set of ordered rules, referred to as the **condition-based classifier (CBC)**. The rules in this set consist of conjunctive terms similar to an associative classifier, but the new rules can contain more conjuncts than the original rules, and the class is determined from the tree. Thus, more expressive models are possible.

Here we introduce one way to construct a CBT and the corresponding CBC. Consider training data set D and a set of associative classification rules generated from D: $\{r_1, r_2, ..., r_J\}$. The procedure of constructing CBT and CBC is shown in Figure 4. Each component in the procedure is introduced

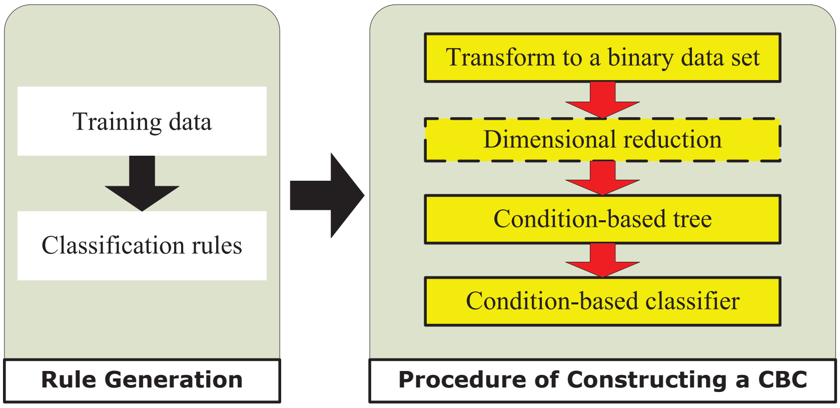

Figure 4: The procedure of constructing a condition-based tree and the corresponding condition-based classifier. After transforming the original data and the classification rules to an indicator data set, feature selection methods can be used for dimensional reduction, and decision tree C4.5 is used to construct a CBT. Then a CBT can be transformed to a CBC, with equivalent classification outcomes. Note the dimensional reduction step is not required, but it can reduce the number of rules in a CBT and the corresponding CBC.

in the following subsections.

3.1. Data Transform

Let $\{c_1, c_2, ..., c_J\}$ denote the condition set of a rule set $\{r_1, r_2, ..., r_J\}$. Let I_{ij} indicate whether a condition $c_j \in \{c_1, c_2, ..., c_J\}$ is satisfied for x_i of a training instance (x_i, y_i). That is,

$$I_{ij} = \begin{cases} 1 & c_j \text{ is satisfied for } x_i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

A new data set I is formed by integrating the indicators and the class labels: $\{[I_{i1}, ..., I_{iJ}, y_i], i = 1, ...N\}$. Let $\{I_1, I_2, ..., I_J\}$ denote the predictor variables. The class variable in I is still Y. Similar data transform techniques were used for a set of rules [2] (where the focus was to organize/prune rules graphically), and a set of patterns [6]. This data step transforms a rule analysis problem to a supervised learning problem, in which I_j $(j = 1, ..., J)$ are predictor variables representing conditions c_j, and Y is the class variable.

As an illustrative example, an exclusive OR (XOR) data set with 160 rows is simulated, and summarized in Table 1. There are only four distinct rows among the 160 rows, we duplicated the rows because an algorithm such as the tree model C4.5 [20] has parameters such as the minimum size for a node. A set of associative classification rules were extracted from the data set (Table 2). The new data set I, shown in Table 3, was formed based on the data in Table 1 and the conditions in Table 2.

data row ID	X_1	X_2	Y
1-40	0	0	0
41-80	0	1	1
81-120	1	0	1
121-160	1	1	0

Table 1: Example data set with 40 rows generated for each XOR logic state.

Rule ID	Conditions	Class
r_1	c_1: {}	Y=0
r_2	c_2: $X_1 = 0$	Y=0
r_3	c_3: $X_2 = 0$	Y=0
r_4	c_4: $X_2 = 1$	Y=0
r_5	c_5: $X_1 = 1$	Y=0
r_6	c_6: $X_1 = 0 \wedge X_2 = 0$	Y=0
r_7	c_7: $X_1 = 1 \wedge X_2 = 1$	Y=0
r_8	c_8: {}	Y=1
r_9	c_9: $X_1 = 0$	Y=1
r_{10}	c_{10}: $X_2 = 0$	Y=1
r_{11}	c_{11}: $X_2 = 1$	Y=1
r_{12}	c_{12}: $X_1 = 1$	Y=1
r_{13}	c_{13}: $X_1 = 0 \wedge X_2 = 1$	Y=1
r_{14}	c_{14}: $X_1 = 1 \wedge X_2 = 0$	Y=1

Table 2: Associative classification rules generated from the XOR data.

data row ID	I_1	I_2	$\cdots$	I_{14}	Y
1-40	1	1	$\cdots$	0	0
41-80	1	1	$\cdots$	0	1
81-120	1	0	$\cdots$	1	1
121-160	1	0	$\cdots$	0	0

Table 3: New data set formed from the raw data in Table 1 and the conditions in Table 2

3.2. Dimensional Reduction via Feature Selection

In the new indicator data set I, an indicator variable I_j corresponds to the rule condition c_j. Therefore, the conditions can be reduced by selecting a feature subset from I. We can apply feature selection (FS) algorithms on I to obtain a reduced subset of conditions and the resultant indicator data set is denoted as I^*. Since feature selection methods are generally developed for high-dimensional data, the condition pruning can be as efficient as the feature selection methods.

The FS algorithm ACE [22] was shown to be generally effective for high-dimensional data and has the advantage that it uses tree-based ensembles. However, a more widely-used method is applied here to focus on the advantages of the CBC with a simpler feature selection method. Consequently, CFS [10] is applied here. For the XOR example, CFS applied to I selects $\{c_6, c_7, c_{13}, c_{14}\}$. The selected conditions correspond to the four XOR logical rules. We also note that the dimensional reduction step is not required, but this can reduce the number of rules in the final classifier, as shown in our experiments discussed later.

$c_7 = 1 \Rightarrow Y = 0$
$c_7 = 0$
$\quad c_6 = 1 \Rightarrow Y = 0$
$\quad c_6 = 0 \Rightarrow Y = 1$

Table 4: The condition-based tree (CBT) for the XOR example.

$X_1 = 1 \wedge X_2 = 1$	$\Rightarrow Y = 0$
$X_1 = 0 \wedge X_2 = 0$	$\Rightarrow Y = 0$
$Else$	$\Rightarrow Y = 1$

Table 5: The condition-based classifier (CBC) extracted from the CBT shown in Table 4.

3.3. Condition-based Tree

A CBT is built by applying an ordinary decision tree (C4.5 for our work) to I or I^*. The decision rule at each node follows: if $I_j = 1$ (if condition c_j is satisfied), then go to one branch; else, go to another branch. The class is assigned at the leaf nodes. The condition-based tree for the XOR example is shown in Table 4. The classifier produces three leaf nodes and correctly captures the true patterns.

3.4. Transform CBT to CBC

A classification rule can be extracted from a CBT via the conjunction of all the condition-value pairs along the path from the root node to a leaf node v. That is, $c_1 = value_1 \wedge c_2 = value_2 \wedge ... \Rightarrow Y = Y_v$, where c_g is a condition in the path, $value_g \in \{True, False\}$, and $Y = Y_v$ is the class assigned at node v.

Each condition-value pair in a rule can be decomposed into a conjunction of attribute-value pairs, but the conditions with value $True$ can be easier to understand than conditions with value $False$. For example, suppose there are three variables $X_1 \in \{small, medium, large\}$, $X_2 \in \{red, green, blue\}$ and $X_3 \in \{square, circle\}$, and two conditions $\{c_1 : X_1 = small \wedge X_2 = green\}$ and $\{c_2 : X_3 = square\}$. Then $c_1 = True$ is equivalent to $X_1 = small \wedge X_2 =$

$green$. In contrast, $c_1 = False$ is equivalent to $X_1 \neq small \vee X_2 \neq green$, and, therefore, equivalent to $X_1 = medium \vee X_1 = large \vee X_2 = green \vee X_2 = blue$, which becomes even more complex when X_1 or X_2 have more categories. Similarly, all conditions with value $True$ can be integrated into a concise conjunction of attribute-value pairs, e.g., $c_1 = True \wedge c_2 = True$ is equivalent to $X_1 = small \wedge X_2 = green \wedge X_3 = square$. However, conditions having value $False$ are more complex and more difficult to execute after transforming to attribute-value pairs. Consequently, we prefer to avoid using the condition-value pairs with value $False$ in a rule. In the following we transform CBT to a set of ordered rules with only condition-value pairs with the value $True$.

At a node, an instance is sent to a child node based on a splitting rule: "instances are sent to the left child node if the condition is true at the node", and "to the right child node if the condition is false". Alternatively, we can present the splitting rules as: "instances are sent to the left child node if the condition is true at the node", and "Else, sent to the right child node". Clearly the later approach avoids using the condition-value pair with the value equal to $False$, but these two splitting rules are ordered. Starting from the root node, we can recursively use this principle to order the rules and avoid using the condition-value pairs with value $False$.

We assume a CBT node stores the following information: $flag\_leaf$ indicates whether the current node is a leaf node, C_{node} is the condition at the current node, $class_{node}$ is the class assigned to a non-leaf node, and $node.child_{left}$ and $node.child_{right}$, respectively, are the left and right child nodes of the current node. Without loss of generality, we assume the data in the left child node of a non-leaf node always satisfies the condition. Algorithm 1 describes a way to transform CBT to an ordered rule set, referred to as CBC. The CBC for the XOR example is shown in Table 5. The CBC correctly and concisely learns the model.

4. Experiments

We used the statistical programming language R [13] and R packages "RWeka" [11] and "arules" [9] to implement the algorithms considered in our experiments. We evaluated our methods on data sets from the UCI Machine Learning Repository [3]. These data sets have been commonly used for evaluating associative classifiers [17, 5] and are summarized in Table 6. To generate associative classification rules, we first discretized the continuous attributes using the entropy method [8] and then extracted the associative

Algorithm 1: $ruleExtract(ruleSet, node, C_{aggregate})$: function to transform a CBT to a set of ordered rule set (CBC).

```
input  : $ruleSet \leftarrow null, node \leftarrow root\ node, C_{aggregate} \leftarrow null$
output: $ruleSet$
if $flag\_leaf = true$ then
    $ruleSet \leftarrow \{ruleSet, C_{aggregate} \Rightarrow class_{node}\}$
end
if $flag\_leaf = false$ then
    $ruleSet \leftarrow ruleExtract(ruleSet, node.child_{left}, C_{aggregate} \wedge C_{node})$
    $ruleSet \leftarrow ruleExtract(ruleSet, node.child_{right}, C_{aggregate})$
end
return $ruleSet$
```

classification rules with minimum support = 0.05 and minimum confidence = 0.6. The sensitivity to the minimum support and confidence was also investigated. In the experiments, we considered CBC-FS (CBC with feature selection), CBC (CBC without feature selection), a decision tree algorithm C4.5 and two associative classifiers CBA and GARC. Here 10-fold cross validation was used to evaluate the algorithms and the results of CBA and GARC were from the original work [17, 5]. To compare CBC-FS to competitors, the Wilcoxon signed-ranks tests were conducted [26].

4.1. Classification Performance

The number of rules of different classifiers are shown in Table 7. Clearly CBC has fewer rules than C4.5, GARC and CBA, and therefore CBC effectively reduced the number of rules used in classification. CBC-FS has even fewer rules than CBC. Consequently, feature selection successfully further reduced the complexity of the classifier here. The Wilcoxon signed ranks tests indicate that CBC-FS has significantly fewer rules than the other methods (at significance level 0.05).

The error rates (%) of different classifiers are also shown in Table 7. According to the Wilcoxon signed ranks tests, CBC-FS is not significantly different from CBC, C4.5 and CBA, but is significantly better than GARC at significance level 0.05. Consequently, feature selection can substantially reduce the number of rules of CBC without significant loss of accuracy performance. This means feature selection is capable of pruning irrelevant or redundant rules.

	Data Summary		
Data Set Name	# Features	# Instances	# Classes
australian	14	690	2
auto	25	205	7
breast	10	699	2
crx	15	690	2
german	20	1000	2
glass	9	214	7
heart	13	270	2
hepatitis	19	155	2
horse	22	368	2
iris	4	150	3
labor	16	57	2
led7	7	3200	10
lymph	18	148	4
pima	8	768	2
tic-tac-toe	9	958	2
vehicle	18	846	4
waveform	21	5000	3
wine	13	178	3
zoo	16	101	7

Table 6: A summary of the data sets used in the experiments.

	Number of rules					Error rates				
Dataset	**CBC-FS**	**CBC**	**C4.5**	**GARC**	**CBA**	**CBC-FS**	**CBC**	**C4.5**	**GARC**	**CBA**
australian	6	13	10	17	148	14.5	14.1	12.9	12.6	13.4
auto	22	23	108	650	54	28.0	27.0	22.0	28.7	27.2
breast	8	12	16	21	49	4.6	5.6	5.3	5.2	4.2
crx	12	27	23	21	142	16.2	16.7	15.7	17.5	14.1
german	16	103	110	78	172	27.3	29.9	28.5	24.8	26.5
glass	14	18	30	17	27	27.6	25.7	25.6	31.9	27.4
heart	7	11	13	12	52	18.1	15.6	18.1	19.4	18.5
hepatitis	5	6	5	23	23	18.1	20.0	21.9	13.3	15.1
horse	4	9	5	26	97	15.0	16.3	15.8	25.0	18.7
iris	2	3	4	7	5	6.7	6.7	6.0	6.0	7.1
labor	4	4	4	15	12	12.3	12.7	19.7	17.7	17.0
led7	27	28	40	33	71	26.4	26.1	25.9	43.5	27.8
lymph	6	8	10	17	36	21.0	20.9	23.7	22.4	19.6
pima	6	7	12	6	45	24.5	24.7	25.1	26.2	27.6
tic-tac-toe	8	8	115	26	8	0.0	0.0	13.8	0.0	0.0
waveform	234	426	692	25	386	21.9	23.2	25.0	28.9	20.6
wine	3	3	12	16	10	8.0	9.1	9.5	16.5	8.4
zoo	7	6	8	90	7	5.0	4.9	5.9	17.7	5.4
mean	**21.7**	**39.8**	**67.7**	**61.1**	**74.7**	**16.39**	**16.61**	**17.80**	**19.84**	**16.59**
median	**6.9**	**10.2**	**12.7**	**21.0**	**47.0**	**17.16**	**16.49**	**18.91**	**18.54**	**17.75**
p-value	**-**	**0.001**	**0.000**	**0.002**	**0.000**	**-**	**0.366**	**0.118**	**0.029**	**1.000**

Table 7: The number of rules used in CBC-FS (CBC with feature selection), CBC (without feature selection), C4.5, GARC and CBA, and the error rates (%) of these classifiers. The mean and median of the number of rules and error rates from each classifier, and the p-values of the Wilcoxon signed ranks tests between CBC-FS and other classifiers are calculated.

We mentioned one key advantage of associative classifiers over ordinary decision trees is that associative classification rules are generated by considering multiple variables simultaneously, while decision trees consider one variable at a time. The tic-tac-toe data set is a perfect example illustrating this point. The tic-tac-toe data set encodes the complete set of possible board configurations at the end of tic-tac-toe games. Let X_1,...,X_9 denote the 9 positions (3 by 3) on a board, and "x" is assumed to have played first. Figure 5 illustrates the tic-tac-toe game board. The target variable Y is "*win*" for 8 possible ways in which there are three "x"s in a row, and is "*lose*" otherwise. Therefore, to capture the patterns of winning a game, a classifier must consider multiple variable simultaneously. C4.5 has more than 100 leaf nodes and still cannot learn all the correct patterns. All the associative classifiers are able to capture the right patterns and have zero error rate. CBC-FS, CBC and CBA have 8 rules, and GARC has 26 rules. The CBC for the tic-tac-toe data set is shown in Table 8.

Also consider the labor data set. The data set includes all collective agreements reached in the business and personal services sector for local unions with at least 500 members (teachers, nurses, university staff, police, etc.) in Canada in 1987 and the first quarter of 1988 [3]. There are 16 attributes including the information about wage increases, employer's contributions to

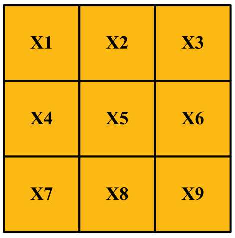

Figure 5: The tic-tac-toe game board. X_1,...,X_9 represent the 9 positions. To win a game, a player must place three chess pieces in a row (e.g., X_1, X_5 and X_9) before the competitor.

$X_3 = x \wedge X_5 = x \wedge X_7 = x$	$\Rightarrow Y = win$
$X_1 = x \wedge X_5 = x \wedge X_9 = x$	$\Rightarrow Y = win$
$X_2 = x \wedge X_5 = x \wedge X_8 = x$	$\Rightarrow Y = win$
$X_1 = x \wedge X_4 = x \wedge X_7 = x$	$\Rightarrow Y = win$
$X_3 = x \wedge X_6 = x \wedge X_9 = x$	$\Rightarrow Y = win$
$X_1 = x \wedge X_2 = x \wedge X_3 = x$	$\Rightarrow Y = win$
$X_7 = x \wedge X_8 = x \wedge X_9 = x$	$\Rightarrow Y = win$
$X_4 = x \wedge X_5 = x \wedge X_6 = x$	$\Rightarrow Y = win$
$Else$	$\Rightarrow Y = lose$

Table 8: The CBC for the tic-tac-toe data set has only 8 rules in addition to the default rule, which correctly captures the pattern for winning a tic-tac-toe game. For this data set, CBA also has 8 rules and GARC has 26 rules, both with zero error rate. C4.5 has more than 100 leaf nodes and does not correctly capture all the patterns.

the pension plan, health plan and dental plan, the number of working hours, education allowance, employer's help during employee long-term disability, etc. The class is either *good* or *bad*, which standards for an acceptable or unacceptable contract, respectively. The CBC for this data set is shown in Table 9. With a compact rule set, CBC is able to achieve lower error rate than other classifiers with more rules.

4.2. Titanic Passenger Survival Data

Here we use CBC to learn what people were likely to survive in the wreck of the Titanic, based on a data set available at Kaggle [15]. The data set consists of 891 passengers and the class is whether a passenger survived (1: survive, 0: non-survive). Particularly, 342 passengers survived and 549 did not. One of the reasons that the shipwreck led to such loss of life was that

$Contribute\ to\ pension\ plan = none$	$\Rightarrow Y = bad$
$Contribute\ to\ health\ plan = none$	$\Rightarrow Y = bad$
$Wage\ increased\ in\ the\ first\ year \leq 2.65$	$\Rightarrow Y = bad$
$Help\ during\ employee\ longterm\ disabil = no$	$\Rightarrow Y = bad$
$Else$	$\Rightarrow Y = good$

Table 9: The CBC for the labor data set. For this data set, GARC has 15 rules and CBA has 12 rules with greater error rates.

there were not enough lifeboats [15]. We considered 7 predictor variables: sex, age, the number of siblings/spouses aboard (sibsp), the number of parents/children aboard (parch), passenger class (1 = 1st; 2 = 2nd; 3 = 3rd), passenger fare and the port of embarkation (C = Cherbourg; Q = Queenstown; S = Southampton). Furthermore, age and fare are discretized into three levels: small, medium and large. We know that, although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class [15]. Therefore, it is interesting to find out if the knowledge from CBC are consistent with human knowledge.

For comparison, a decision tree C4.5 was applied to the data, and the rules from the tree are shown in Table 10. The two numbers (n_1/n_2) after each rule are, respectively, are the number of passengers satisfying the condition (the left hand side of the rule), and the number of passengers with a different outcome from the rule consequent. For example, for the first rule, there were totally 94 passengers satisfying "$sex = female \wedge pclass = 1$", and 3 of them did not satisfy $Y = 1$. A rule with a smaller n_2 (given the same n_1) indicates a smaller number of exceptions, and greater accuracy.

$sex = female \wedge pclass = 1$	$\Rightarrow Y = 1$ (94/3)
$sex = female \wedge pclass = 2$	$\Rightarrow Y = 1$ (76/6)
$sex = female \wedge pclass = 3 \wedge embarked = C$	$\Rightarrow Y = 1$ (23/8)
$sex = female \wedge pclass = 3 \wedge embarked = Q$	$\Rightarrow Y = 1$ (33/9)
$sex = female \wedge pclass = 3 \wedge embarked = S$	$\Rightarrow Y = 0$ (88/33)
$sex = male$	$\Rightarrow Y = 0$ (577/109)

Table 10: The decision tree for the Titanic data set.

The decision tree reveals that female passengers were more likely to survive than male passengers, but misses other information. There is only one

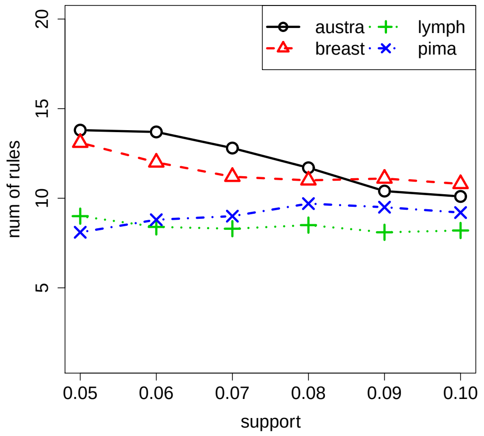

Figure 6: The number of rules in CBC as the minimum support changes with fixed minimum confidence = 0.6.

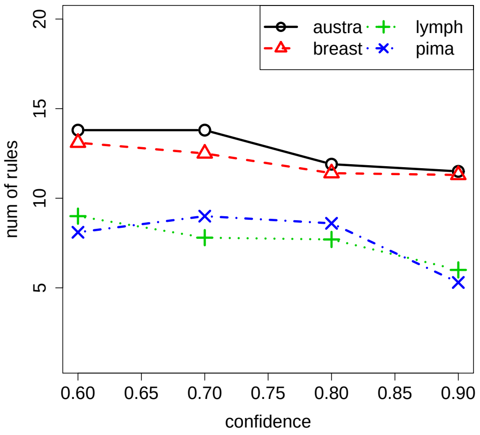

Figure 7: The number of rules for CBC as the minimum confidence changes with fixed minimum support = 0.05.

rule describing the male passengers and that rule has 109 exceptions and is not accurate. In fact, the training error rate from the tree is only 0.047 for class 0, but is as large as 0.42 for class 1. Therefore, many passengers survived are classified to 0 (not-survived).

The CBC with minimum support=0.05 and confidence=0.5 for this data set is shown in Table 11. Although CBC has more rules than the decision tree, these rules are not irrelevant or redundant. The training error rate from CBC is 0.07 for class 0, and is 0.27 for class 1, significantly lower than decision tree (0.42), which indicates that these rules capture useful information about class 1. Some CBC rules have three or more attribute-value pairs in their conditions and are reasonably accurate, while the decision tree contains rules with at most three attribute-value pairs. For both CBC and decision trees, rules with long conditions are expected to have high confidence, otherwise the conditions can be pruned to maintain the trade-off between accuracy and complexity. Therefore, this demonstrates that CBC is more capable of discovering highly-confident rules.

Also note this is an imbalanced classification problem. That is, there are more passengers who did not survive. It can be seen that C4.5 has unsatisfactory performance on identifying survivors, while CBC can capture more information for predicting survivors.

$pclass = 1 \wedge sex = female$	$\Rightarrow Y = 1$ (94.0/3.0)
$pclass = 2 \wedge sex = female$	$\Rightarrow Y = 1$ (76.0/6.0)
$sex = female \wedge sibsp = 0 \wedge age = large$	$\Rightarrow Y = 0$ (7.0/1.0)
$sex = female \wedge sibsp = 0 \wedge parch = 0 \wedge fare = medium$	$\Rightarrow Y = 0$ (10.0/4.0)
$sex = female \wedge sibsp = 0 \wedge fare = small \wedge embarked = S \wedge age = small$	$\Rightarrow Y = 1$ (7.0/2.0)
$sex = female \wedge sibsp = 0 \wedge fare = small \wedge embarked = S$	$\Rightarrow Y = 0$ (11.0/4.0)
$sex = female \wedge sibsp = 0$	$\Rightarrow Y = 1$ (46.0/12.0)
$sex = female \wedge embarked = C \wedge sibsp >= 2$	$\Rightarrow Y = 1$ (3.0)
$sex = female \wedge embarked = C \wedge age = small \wedge parch = 0$	$\Rightarrow Y = 1$ (3.0/1.0)
$sex = female \wedge embarked = C$	$\Rightarrow Y = 0$ (4.0/1.0)
$sex = female \wedge parch = 0 \wedge fare = medium \wedge age = small$	$\Rightarrow Y = 0$ (4.0)
$sex = female \wedge parch = 0 \wedge fare = medium$	$\Rightarrow Y = 1$ (14.0/4.0)
$pclass = 1 \wedge sibsp = 1 \wedge age = small \wedge parch = 0$	$\Rightarrow Y = 0$ (2.0)
$pclass = 1 \wedge sibsp = 1 \wedge age = small$	$\Rightarrow Y = 1$ (2.0)
$pclass = 1 \wedge sibsp = 1 \wedge parch = 0 \wedge embarked = C$	$\Rightarrow Y = 1$ (7.0/2.0)
$pclass = 1 \wedge sibsp = 1$	$\Rightarrow Y = 0$ (20.0/8.0)
$sibsp = 0 \wedge fare = large \wedge parch >= 2 \wedge embarked = S$	$\Rightarrow Y = 1$ (2.0)
$sibsp = 0 \wedge fare = large \wedge age = medium \wedge parch = 0$	$\Rightarrow Y = 1$ (16.0/6.0)
$sibsp = 0 \wedge fare = large$	$\Rightarrow Y = 0$ (72.0/24.0)
$parch = 1 \wedge age = small \wedge sibsp = 0$	$\Rightarrow Y = 1$ (2.0)
$parch = 1 \wedge sibsp = 1 \wedge fare = medium \wedge age = medium$	$\Rightarrow Y = 0$ (7.0/1.0)
$parch = 1 \wedge sibsp = 1 \wedge fare = medium \wedge sex = male \wedge age = large$	$\Rightarrow Y = 0$ (2.0)
$parch = 1 \wedge sibsp = 1 \wedge fare = medium$	$\Rightarrow Y = 1$ (16.0/2.0)
$parch = 1$	$\Rightarrow Y = 0$ (29.0/2.0)
$Else$	$\Rightarrow Y = 0$ (435.0/47.0)

Table 11: The CBC for the Titanic data set. The rules are ordered by priority, and should be executed from the top to the bottom.

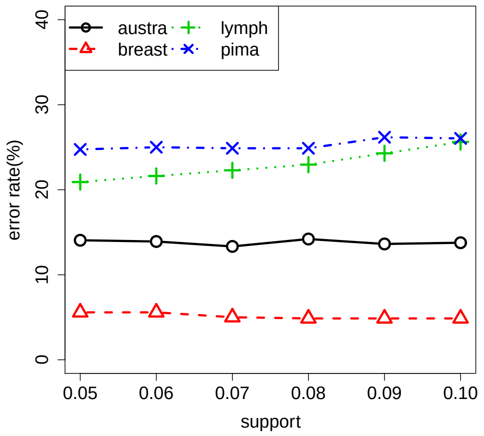

Figure 8: The error rates of CBC as the minimum support changes with fixed minimum confidence = 0.6.

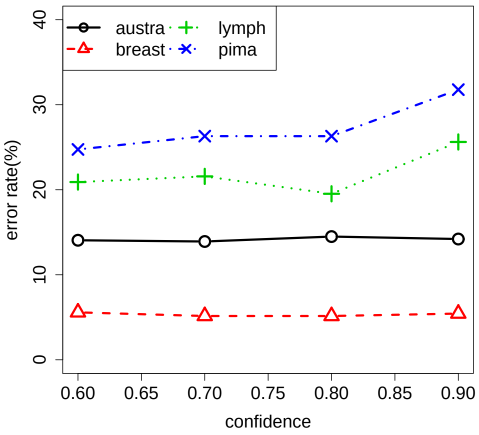

Figure 9: The error rates of CBC as the minimum confidence changes with fixed minimum support = 0.05.

4.3. Sensitivity to Support and Confidence

Here we investigate the sensitivity of CBC to the changes of minimum support and minimum confidence for mining associative classification rules. First we fix the minimum support = 0.05 and analyze different minimum confidence values: {0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9}. Then we fix the minimum confidence = 0.6 and analyze different minimum support values: { 0.05, 0.06, 0.07, 0.08, 0.09, 0.1}. The number of rules or error rates versus the minimum support or minimum confidence, for four data sets (other data sets are omitted to avoid too many lines on the figures), are shown in Figure 6, Figure 7, Figure 8 and Figure 9.

As expected, for most data sets, the number of rules tends to decrease as the minimum support or minimum confidence increases (shown in Figure 6 and Figure 7). When the thresholds increase, the number of rules that can be used decreases, and, thus, the classifiers become more concise. Because some useful rules can be eliminated by setting a larger threshold, the error rates tend to increase for data sets *lymph* and *pima* (shown in Figure 8 and Figure 9). Also, for these two data sets, the decrease in the number of rules in CBC is relatively large when changing the minimum confidence from 0.8 to 0.9, and, correspondingly, the increase in error rate is also relatively large for the same change. This indicates that the rules with confidence greater than 0.8 provide the main information for building the CBC, and removing these rules has a negative impact on the accuracy. In contrast, the error rate changes for the other two data sets *austra* and *breast* are not obvious even when increasing minimum confidence to 0.9. This indicates that only the rules with confidence greater than 0.9 are needed to capture the characteristics of the data sets, which may also indicate that the classification problems are less complex. Indeed, the error rates of the two data sets are smaller than data sets *lymph* and *pima*. Furthermore, from experiments not shown here, we found that the change in the average error rate over all the data sets is reasonably stable with regard to minimum support or minimum confidence, while the average number of rules tends to decrease as the minimum support or minimum confidence increases.

5. Conclusions

Ordinary decision trees consider one variable at a time, and, thus, they are challenged by problems where the interactions of multiple variables are predictive. We illustrate this by two examples, the exclusive OR logic and the tic-tac-toe data set. Associative classification leverages association rules which consider multiple variables simultaneously and, therefore, can capture variable interactions well. However, when an associative classifier consisting of an ordered rule set (e.g., CBA) is represented as a tree, at least one child node of a non-leaf node in the tree is never split further. The expressiveness of the tree may be limited.

We propose the condition-based tree (CBT) that relaxes the restrictions of both ordinary trees and associative classifiers. To facilitate interpretability, we propose a condition-based classifier (CBC), which is a equivalent representation of a CBT, but with simplified conditions. Experimental studies show that CBC has significantly fewer rules than existing associative classifiers: CBA and GARC, with similar accuracy.

We also found feature selection can substantially reduce the number of rules in CBC. CBC is built using a decision tree algorithm. [7] discussed the node sparsity issue in decision tree algorithms. That is, many features can share the same information gain if there are a small number of instances and a large number of features in a node. Thus, a feature that is not globally strong can be used to split the node. Feature selection scores the features using all the data and may be expected to select strongly relevant features, which prevents a decision tree from splitting on "noisy" features and becoming unnecessarily large.

Existing associative classifiers can be accurate, however, they often have a large number of rules. Having a compact set of rules in a rule-based classifier is a key for decision makers to understand, validate and trust the data-driven model. Therefore, this work promotes associative classification as an accurate and comprehensible data-driven model.

Although CBC or CBC-FS, in general, has fewer rules than the competitors with similar accuracy, in some cases such as the Titanic passenger survival prediction problem, CBC has more rules than an ordinary decision tree C4.5. This may be because CBC is able to form rules with high confidence, and, thus, rules are not pruned. As expected, in the Titanic passenger survival prediction problem, C4.5 misses some important rules and has unsatisfactory performance on identifying survivals.

Because CBC uses the combination of multiple attributes to split a node, individual rules in CBC can be longer than rules in ordinary trees that only use one attribute at a node. However, the complexity of individual rules depends on the nature of the data set. The length of rules is expected to increase as the complexity of the data set increases. For example, the maximum number of attributes in a rule for the XOR example is two, while the maximum number of attributes in a rule for the tic-tac-toe example is three.

CBC inherits some favorable characteristics from associative rule algorithms and decision trees. For example, the punning function is able to prevent over-fitting and maintain a balance between training accuracy and model complexity. On the other hand CBC also has a few disadvantages similar to associative rule algorithms. First, the attributes have to be categorical or need to be discretized. Discretization may cause information lost, and, therefore, might have a negative impact on accuracy. Second, associative rules are often generated in a brute-force mode, and, therefore, the efficiency becomes a concern for high-dimensional data. To solve this issue, one can perform feature selection to reduce the number of attributes. Also, one can limit the length of rule conditions to avoid expensive computations.

It should be noted that the classification rules used to construct CBT do not have to be associative classification rules. Future research includes building CBT based on rules/conditions from efficient algorithms such as tree ensembles, which naturally handle mixed numerical and categorical variables.

It may also be of interest to apply the CBT-CBC rule transformation algorithm to binary decision trees. The rules after transformation should be executed in a certain order, but can have much simpler conditions than the rules generated by the ordinary way, that is, extracting a rule by aggregating the variable-value pairs from the root node to each leaf node.

Acknowledgments

We thank the anonymous reviewers for the insightful comments.

References

[1] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: Proceedings of the 20th International Conference on Very Large Data Bases, Morgan Kaufmann, 1994, pp. 487–499.

[2] A. Berrado, G. Runger, Using metarules to organize and group discovered association rules, Data Mining and Knowledge Discovery 14 (2007) 409–431.

[3] C. Blake, C. Merz, UCI repository of machine learning databases (1998).

[4] L. Breiman, J. Friedman, R. Olshen, C. Stone., Classification and Regression Trees, Wadsworth, Belmont, MA, 1984.

[5] G. Chen, H. Liu, L. Yu, Q. Wei, X. Zhang, A new approach to classification based on association rule mining, Decision Support Systems 42 (2006) 674–689.

[6] H. Cheng, X. Yan, J. Han, C. Hsu, Discriminative frequent pattern analysis for effective classification, in: Proccedings of the 23rd International Conference on Data Engineering, IEEE, 2007, pp. 716–725.

[7] H. Deng, G. Runger, Gene selection with guided regularized random forest, Pattern Recognition 46 (2013) 3483–3489.

[8] U.M. Fayyad, K.B. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, in: Proceedings of the 13th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1993, pp. 1022–1027.

[9] M. Hahsler, B. Gruen, K. Hornik, A computational environment for mining association rules and frequent item sets, Journal of Statistical Software 14 (2005) 1–25.

[10] M.A. Hall, Correlation-based feature selection for discrete and numeric class machine learning, in: Proceedings of the 17th International Conference on Machine Learning, Morgan Kaufmann, 2000, pp. 359–366.

[11] K. Hornik, C. Buchta, A. Zeileis, Open-source machine learning: R meets Weka, Computational Statistics 24 (2009) 225–232.

[12] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, B. Baesens, An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models, Decision Support Systems 51 (2011) 141–154.

[13] R. Ihaka, R. Gentleman, R: A language for data analysis and graphics, Journal of computational and graphical statistics 5 (1996) 299–314.

[14] Y. Jiang, J. Shang, Y. Liu, Maximizing customer satisfaction through an online recommendation system: A novel associative classification model, Decision Support Systems 48 (2010) 470–479.

[15] Kaggle, Titanic: Machine learning from disaster, 2013. URL: `http://www.kaggle.com/c/titanic-gettingStarted/data`.

[16] W. Li, J. Han, J. Pei, Cmar: accurate and efficient classification based on multiple class-association rules, in: Proceedings of the 2001 IEEE International Conference on Data Mining, IEEE, 2001, pp. 369–376.

[17] B. Liu, W. Hsu, Y. Ma, Integrating classification and association rule mining, in: Proceeding of the 1998 International Conference on Knowledge Discovery and Data Mining, ACM, 1998, pp. 80–86.

[18] D. Martens, J. Vanthienen, W. Verbeke, B. Baesens, Performance of classification models from a user perspective, Decision Support Systems 51 (2011) 782–793.

[19] D. Papakiriakopoulos, K. Pramatari, G. Doukidis, A decision support system for detecting products missing from the shelf based on heuristic rules, Decision Support Systems 46 (2009) 685–694.

[20] J. Quinlan, C4.5: programs for machine learning, Morgan Kaufmann, 1993.

[21] P. Su, W. Mao, D. Zeng, H. Zhao, Mining actionable behavioral rules, Decision Support Systems 54 (2012) 142–152.

[22] E. Tuv, A. Borisov, G. Runger, K. Torkkola, Feature selection with ensembles, artificial variables, and redundancy elimination, Journal of Machine Learning Research 10 (2009) 1341–1366.

[23] A. Veloso, W. Meira, M. Zaki, Lazy associative classification, in: Proceeding of the 6th International Conference on Data Mining, IEEE, 2006, pp. 645–654.

[24] J. Wang, G. Karypis, HARMONY: Efficiently mining the best rules for classification, in: Proceedings of the 2005 SIAM International Conference on Data Mining, SIAM, 2005, pp. 205–215.

[25] K. Wang, S. Zhou, Y. He, Growing decision trees on support-less association rules, in: Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining, ACM, pp. 265–269.

[26] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (1945) 80–83.

[27] J. Yeh, T. Wu, C. Tsao, Using data mining techniques to predict hospitalization of hemodialysis patients, Decision Support Systems 50 (2011) 439–448.

[28] X. Yin, J. Han., Cpar: Classification based on predictive association rules, in: Proceedings the 2003 SIAM International Conference on Data Mining, SIAM, 2003, pp. 331–335.

[29] O. Zaïane, M. Antonie, On pruning and tuning rules for associative classifiers, in: Knowledge-Based Intelligent Information and Engineering Systems, Springer, 2005, pp. 966–973.