Chapter 10
Irrelevant Feature and Rule Removal for Structural Associative Classification Using Structure-Preserving Flat Representation

Izwan Nizal Mohd Shaharanee and Fedja Hadzic

Abstract Practical applications of association rule mining often suffer from overwhelming number of rules that are generated, many of which are not interesting or useful for the application in question. Removing irrelevant features and/or rules comprised of irrelevant features can significantly improve the overall performance. Many statistical and constraint based measures are used to discard unnecessary and irrelevant features and rules when vectorial or tabular data is in question. In contrast, the use of such measures is limited in the tree-structured data domain, due to the structural aspects that are not easily incorporated. In this chapter, we explore the use of a feature subset selection measure as well as a number of common statistical interestingness measures via a recently proposed structure-preserving flat representation for tree-structured data such as XML. A feature subset selection is used prior to association rule generation. Once the initial set of rules is obtained, irrelevant rules are determined as those that are comprised of attributes not determined to be statistically significant for the classification task. The experiments are performed using real world web access trees and property management dataset. The results indicate that where the dataset has more standard structure a large number of insignificant rules will be discarded and accuracy will increase. However, where the tree instances can vary greatly in terms of structure and label distribution among nodes, while many rules are removed and the accuracy increases, there is a significant reduction in coverage rate of the rule set.

Keywords Tree-structured data · Association rule based classification · Feature subset selection · Statistical interestingness

I.N.M. Shaharanee (✉)
School of Quantitative Sciences, Universiti Utara Malaysia, Sintok, Malaysia
e-mail: nizal@uum.edu.my

F. Hadzic
Department of Computing, Curtin University, Perth, Australia
e-mail: fedja.hadzic@curtin.edu.au

U. Stańczyk and L.C. Jain (eds.), *Feature Selection for Data and Pattern Recognition*, Studies in Computational Intelligence 584,
DOI 10.1007/978-3-662-45620-0_10

10.1 Introduction

Real world datasets often contain attributes that are irrelevant or redundant for the classification problem at hand. These features can degrade the performance and interfere with the learning mechanism typically resulting in a reduction in the quality and generality of the discovered patterns/model and overfitting of the model to the train data. The basic principle of feature subset selection is to find the necessary and sufficient subset of features or attributes which results in simplification of the discovered knowledge model, better generalisation power, while at the same time the accuracy for classification tasks is not compromised.

Association rule mining, being one of the most popular techniques for discovering interesting associations among data objects, has also been utilized for the classification task, where it can contribute to discovering strong associations between occurring attribute and class values [26]. An associative classification framework was first proposed in [28], which consists of an algorithm to generate all class association rules from which a classifier is constructed. Many works [10, 45, 49, 50] have developed various extensions and refinements to this initially proposed framework and the results reported high accuracy and efficiency for the classification problem. Similarly in tree-structured data domain, the XRules structural classifier [52], is based on association rules generated from the ordered embedded subtree mining algorithm [51].

When dealing with pattern selection, one faces the quantity problem due to large volume of output as well as the quality assurance problem of rules reflecting real, significant associations in the domain under investigation [25]. In a recent work presented in [24] the search space of Apriori-like algorithms is pruned so that discovered rules are interesting with respect to the Jaccard measure, rather than the support constraint for which an optimal threshold is often unknown. To deal with the quality problem many interestingness measures have been developed and utilized in various knowledge discovery tasks [12, 29]. In one train of thought, since the nature of data mining techniques is data-driven, the generated rules can often be effectively validated by a statistical methodology in order for them to be useful in practice [13, 22]. Interesting rules could then be interpreted as those rules that have a sound statistical basis and are neither redundant nor contradictory. The aforementioned works [12, 13, 22, 29] have mainly focused on relational data. There is relatively less work in this area when it comes to tree-structured data (an overview is given in the next section). Tree-structured data has underlying complex structural characteristics which typically need to be preserved in the knowledge patterns discovered during a data mining task [17, 52]. The structural characteristics of data pose difficulties in application of traditional classifiers and interestingness measures, whose mechanism typically does not take structural aspects of data into account.

In [38], a unified framework was proposed that systematically combines several statistical/heuristic techniques to assess the rule quality and remove any redundant and unnecessary rules for the classification problem. In this chapter, the focus is on

exploring the application of this framework to tree-structured data, enabled by the recently proposed structure-preserving flat data format for tree-structured data [14]. The work presented in [14] is based on the extraction of a *database structure model* (henceforth DSM) within which every tree instance from the database can be matched to and which keeps the structural information of the flat representation generated. The implications of the representation in contrast to traditional tree mining field is that every subtree pattern or a rule, will be constrained by the pre-order position of the constituting tree nodes of the subtree w.r.t the DSM. In this work, we explore the application of a feature subset selection measure and statistical interestingness measures via this method to filter out unnecessary and irrelevant subtree patterns for the structural classification task. A feature subset selection method is used prior to association rule generation. Once the initial set of rules is obtained, irrelevant rules are determined as those that are comprised of attributes not determined to be statistically significant for the classification task. The experiments are performed using real world web access tree dataset and a property management dataset from a real estate company. The results indicate that where the dataset has more standard structure the use of statistical measures will discard a large number of insignificant rules and at the same time increase the accuracy of the rule sets. On the other extreme, where the tree instances can vary greatly in terms of structure and label distribution among nodes, as is the case in the web access tree dataset, while many rules are removed and the accuracy increases, there is a significant reduction in coverage rate of the rule set. Furthermore, we compare some of the results with a structural classifier based on traditional subtrees, and highlight some important differences and implications when subtree based rules are constrained by their position. The results also show that including the associations that do not necessary result in connected trees can be important, while such associations are typically ignored within the tree mining field. These findings indicate that structural classifier could be improved and complemented by including disconnected subtrees and constraining the subtrees by their exact occurrence in the database. However, more work is required to identify the domains and application where including such association rules can be beneficial and the right way to combine them with traditional subtree patterns for optimal performance.

The rest of the chapter is organized as follows. The related works are given in Sect. 10.2, while Sect. 10.3 defines the concepts and the rule set optimization problem focused on in this study. In Sect. 10.4, we describe the steps involved in the proposed approach which is evaluated using real-world datasets and experimental findings are discussed in Sect. 10.5. Section 10.6 concludes the chapter.

10.2 Related Work

To date, limited work has been done on the feature selection, rule evaluation and interestingness measures for tree-structured data. Many of the well developed rule interestingness measures are in relational data and they have had great success in

evaluating rule interestingness as discussed in [44]. Several works on the evaluation of discovered patterns based on statistical significance [2, 22, 46] are limited to relational data. The existence of vast well-developed measuring techniques to evaluate interestingness of rules from relational data, offers great opportunities for adapting these techniques for verifying significant subtrees from semi-structured data. The applicability of these interestingness measures needs to be explored in the context of frequent subtree mining, where necessary adjustments and extensions need to be made to ascertain the validity of the methods given the more complex structured aspects in the data, which often need to be preserved in the rules.

One line of work focusing on more interesting subtree patterns aims to reduce the patterns through the application of plausible constraints. The problem of mining mutually dependent ordered subtrees has been addressed in [32]. The proposed algorithm utilizes the hyperclique method [47] in the tree mining context so that all the components of a subtree are highly correlated together. These hyperclique subtree patterns are discovered using an h-confidence measure which is the minimum probability of an item from a pattern in one transaction implying the presence of all other items in the same transaction. Hence, the extracted hyperclique subtree patterns will satisfy the minimum h-confidence threshold. The work done in [3] uses the method proposed for database compression in regards to item set mining in [39] to demonstrate how the same minimum description length principle can yield good results for sequential and tree-structured data. The work presented in [31] extends the idea of the item constraint [41] to that of a node-inclusion constraint in subtrees. Furthermore, Knijf and Feelders [20] proposed the use of monotone constraints in frequent subtree mining, namely monotone, anti-monotone, convertible and succinct constraints. Using these constraints, the frequent subtrees are mined using an opportunistic pruning strategy, and the set of frequent subtrees are reduced to only those satisfying the specific user pre-defined constraints.

Besides the aforementioned constraint-based techniques, to the best of our knowledge, there are limited works on verifying the significance of discovered frequent subtrees. Hashimoto et al. [19] proposed and developed an application of statistical hypothesis testing to re-rank the significant frequent subtrees. This approach ranks the significant patterns according to P-values obtained from the Fisher's Exact test of significance. The significant patterns were then used for Glycan classifications problems. Recently Yan et al. [48], proposed a mining framework called LEAP (Descending Leap Mine) for checking and mining significant frequent subgraphs which helps to discard redundant frequent subgraphs. For a predefined class label in XML documents, an efficient XRules classifier has been proposed in [52]. This approach offers promising results in terms of a structural classifier for semi-structured data, but utilizes standard measures of interestingness based on support and confidence.

10.2.1 Relationship Between Feature Subset Selection and Frequent Subtree Interestingness

In general, the objective of feature subset selection as defined in [18] is to find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes. Han and Kamber in [18] asserted that domain expertise can be employed in order to pick out useful attributes. However, because the data mining task involves a large volume of data and unpredictable behaviour of data during data mining, this task is often expensive and time consuming.

The test of statistical significance is one of the prominent approaches in evaluating attributes/features usefulness. Stepwise forward selection, stepwise backward selection and a combination of both are three commonly used heuristic techniques utilized in statistical significance tests such as linear regression and logistic regression [18]. Moreover, the application of correlation analysis such as the chi-square test is also valuable in identifying redundant variables for feature subset selection. Another powerful technique for this purpose is the Symmetrical Tau [54], which is a statistical-heuristic feature selection criterion. It measures the capability of an attribute in predicting the class of another attribute. Additionally, information gain is another attributes relevance analysis method employed in the popular ID3 [33] and C4.5 [34] as reported in [18]. An extensive overview and comparison of the different approaches to the feature subset selection problem has been provided in [6, 11, 21, 30].

While the original purpose of feature subset selection is to reduce the number of attributes to only those attributes relevant for a certain data mining task, they nevertheless can be utilized to measure the interestingness of rules/pattern generated. For example, if the rule/pattern consists of irrelevant attributes, the aforementioned measure can also give some indication that the rule/pattern is not interesting. Moreover, [12] stated that there are three roles of interestingness measures. The first is their ability to discard uninteresting patterns during the mining process, thereby narrowing the search space and improving the mining efficiency. The second role is to calculate the interestingness scores for each pattern, which allows the ranking of patterns according to specific needs. The final role is the use of interestingness measures during the post-processing stage to select interesting patterns. Interestingness measures such as the chi-square test [8], Symmetrical Tau [54] and Mutual Information [44], are capable of measuring the interestingness of rules and at the same time identifying useful features for frequent patterns.

Since frequent patterns are generated based solely on frequency without considering their predictive power, the use of frequent patterns without selecting appropriate features will still result in a huge feature space which leads to larger volume and complexity of rules. This might not only slow down the model learning process, but even worse, the classification accuracy deteriorates (another kind of overfitting issue since the features are numerous) [9].

10.3 Problem Background

The problem of finding association rules $x \rightarrow y$ was first introduced in [1] as a data mining task for finding frequently co-occurring items in large databases. Let $I = \{i_1, i_2, \ldots, i_{|I|}\}$ be a set of items. Let D be a transactions database for which each record/transaction R is a set of items, such that $R \subseteq I$. An association rule is an implication of the form $x \rightarrow y$ where $x \subseteq I$ and $y \subseteq I$ and $x \cap y = \emptyset$. The absolute support of a rule $x \rightarrow y$ is the number of transactions that contain both x and y. Typically, the relative support is used, where given the support of rule $x \rightarrow y$ (denoted as $\sigma\ (x \rightarrow y)$) be $s\,\%$, there are $s\,\%$ of transactions in D that contain items (itemsets) x and y. In other words, the probability $P(x \cup y) = s\,\%$. An itemset is frequent if it satisfies the user-specified minimum support threshold. The confidence of a rule $x \rightarrow y$, is the estimate of conditional probability of a transaction containing the consequent (y) if the transaction contains the antecedent (x), and is calculated as $\sigma(x \rightarrow y)/\sigma(x)$.

Association rule discovery finds all rules that satisfy specific constraints such as the minimum support and confidence threshold, as is the case in the Apriori algorithm [1]. When tree-structured data such as XML is in question, the underlying associations are tree-structured by nature. Thus, the pre-requisite for the discovery of (structural) association rules becomes the task of frequent subtree mining. A tree-structured document can be modeled as a rooted ordered labeled tree. A *rooted ordered labeled tree* can be denoted as $T = (v_0, V, L, E)$, where (1) $V_0 \in V$ is the root vertex; (2) V is the set of vertices or nodes; (3) L is a labelling function that assigns a label $L(v)$ to every vertex $v \in V$; (4) $E = \{(v_1, v_2) | v_1, v_2 \in V$ AND $v_1 \neq v_2\}$ is the set of edges in the tree, and (5) for each internal nodes, the children are ordered from left to right.

This problem is generally defined as: given a database of trees T_{db} and minimum support threshold σ, find all subtrees that occur at least σ times in T_{db}. Most commonly considered subtrees are induced and embedded. The formal definitions of induced and embedded subtrees are as follows [42]: Given a tree $S = (vs_0, V_S, L_S, E_S)$ and tree $T = (vt_0, V_T, L_T, E_T)$, S is an ordered *induced* subtree of T iff (1) $V_S \subseteq V_T$; (2) $L_S \subseteq L_T$, and $L_S(v) = L_T(v)$; (3) $E_S \subseteq E_T$; (4) the left-to-right ordering of sibling nodes in the original tree is preserved. Moreover, S is an ordered *embedded* subtree of T iff (1) $V_S \subseteq V_T$; (2) $L_S \subseteq L_T$, and $L_S(v) = L_T(v)$; (3) if $(v_1, v_2) \in E_S$ then $parent(v_2) = v_1$ in S and v_1 is ancestor of v_2 in T, and (4) the left-to-right ordering of sibling nodes in the original tree is preserved. If $S = (vs_0, V_S, L_S, E_S)$ is an embedded subtree of $T = (vt_0, V_T, L_T, E_T)$, and two vertices $v_1 \in V_S$ and $v_2 \in V_S$ form ancestor-descendant relationship, the *level of embedding* (LoE) [42], between v_1 and v_2, denoted by $\triangle\ (v_1, v_2)$, is defined as the length of the path between v_1 and v_2 in T. Hence, a *maximum level of embedding constraint* (MaxLoE) $M_\triangle$ can be imposed on the subtrees extracted from T, such that any two connected nodes in an embedded subtree of T will be connected in T by a path that has the maximum length of $M_\triangle$. Examples of induced and embedded subtree are given in Fig. 10.1 (the number on the left of the nodes indicate its pre-order position in the original tree T).

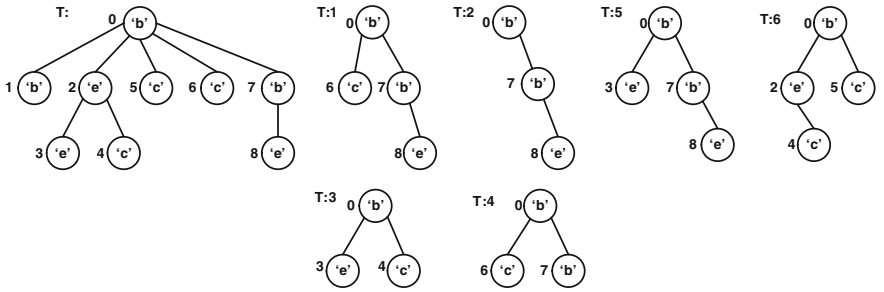

Fig. 10.1 Example of induced/embedded subtrees (T_1, T_2, T_4, T_6) and embedded subtrees (T_3, T_5) of tree T

In this chapter, the focus is on evaluating rules based on embedded and induced subtrees that satisfy minimum support and confidence thresholds, and discarding any rules determined to be irrelevant to the classification task at hand. Let us denote the subtree patterns from the frequent subtree set SF that have a class label (value), as SFC, their accuracy as $ac(SFC)$ and coverage rate as $cr(SFC)$. The problem focused on in this work can then be generally defined as follows: Given SFC with accuracy $ac(SFC)$, obtain $SFC' \subseteq SFC$, such that $ac(SFC') \geq (ac(SFC) - \varepsilon)$ and $cr(SFC') \geq (cr(SFC) - \varepsilon)$ (ε is an arbitrary user defined small value used to reflect the noise that is often present in real-world data).

In what follows we discuss the common way of representing trees. This will lay the necessary ground for understanding the positional constraint imposed by the DSM approach [14]. A pre-order traversal can be computed as follows: If an ordered tree T consists only of a root node r, then r is the pre-order traversal of T. Otherwise let $T_1, T_2, \ldots, T_n$ be the subtrees occurring at r from left to right in T. The pre-order traversal begins by visiting r and then traversing all the remaining subtrees in pre-order starting from T_1 and finishing with T_n. The string encoding (φ) can be generated by adding vertex labels in the pre-order traversal of a tree $T = (v_0, V, L, E)$ and appending a backtrack symbol (e.g., '/', '/' $\notin L$) whenever we backtrack from a child node to its parent node. Figure 10.2 and Table 10.1 depict a tree database consisting of 7 tree instances (or transactions) and the string encoding for tree database, respectively.

10.3.1 Feature Subset Selection

Feature subset selection is an important pre-processing step in the data mining process. If the irrelevant attributes are left in the dataset, they can interfere with the data mining process and the quality of the discovered patterns may deteriorate, creating problems such as overfitting [9]. It is in particular the case in associative classifiers, since frequent patterns are typically generated without considering their predictive power [9], resulting in a huge feature space for possible frequent patterns. The removal of irrelevant attributes will result in a much smaller dataset, thereby

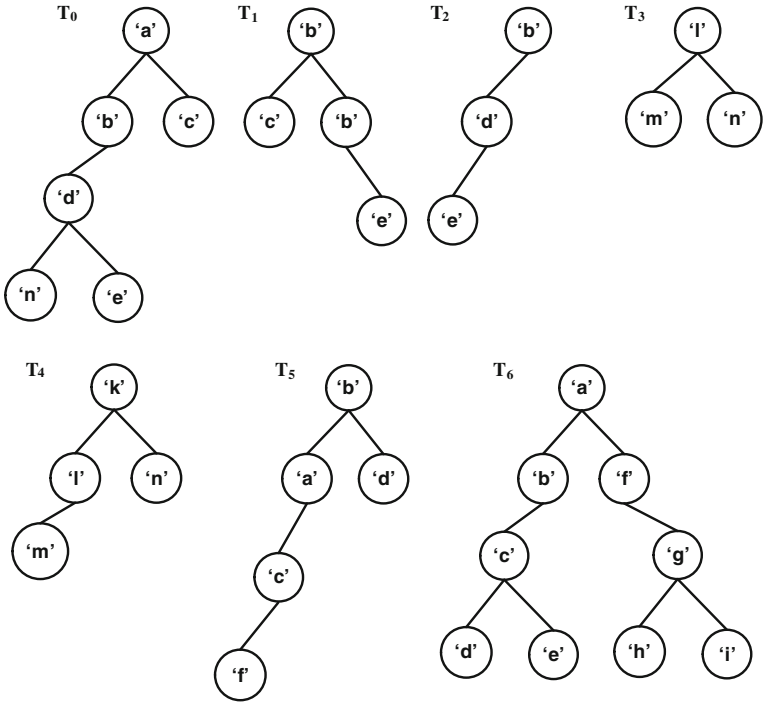

Fig. 10.2 Example of a tree-structured database T_{db} consisting of 7 transactions

Table 10.1 Example of tree transactions

Tree database (T_{db})	Pre-order string encoding
T_0	'a b d n −1 e −1 −1 −1 c −1'
T_1	'b c −1 b e −1 −1'
T_2	'b d e −1 −1'
T_3	'l m −1 n −1'
T_4	'k l m −1 −1 n −1'
T_5	'b a c f −1 −1 −1 d −1'
T_6	'a b c d −1 e −1 −1 −1 f g h −1 i −1 −1 −1'

reducing the number of rules that need to be generated from the association rule mining algorithm, while closely maintaining the integrity of the original data [18]. Additionally, rules described with fewer attributes are also expected to perform better when classifying future cases; hence, they will have better generalization power than do the more specific rules that take many attributes into account. Besides, the patterns extracted will also be simpler and easier to analyse and understand. Determining the relevant and irrelevant attributes poses a great challenge to many data mining algorithms [36].

The feature subset selection problem can be more formally described as: Given a relational database D, $AT = \{at_1, at_2, \ldots, at_{|AT|}\}$ the set of distinct items in D, and $Y = \{y_1, y_2, \ldots, y_{|Y|}\}$ the class attribute with a set of class labels in D. Let an association rule mining algorithm be denoted as AR_{AL}, the set of association rules for predicting the value of a class attribute Y from D extracted using AR_{AL} as $AR(D)$, and accuracy of $AR(D)$ as $ac(AR(D))$. The problem of feature subset selection is to reduce D into D' such that $AT' \subseteq AT$ and $ac(AR(D')) \geq ac(AR(D')) - \varepsilon$, where ε is an arbitrary user defined small value to reflect noise present in real-world data. In other words, the task is to find the optimal set of attributes, $AT_{OPT} \subseteq AT$, such that the accuracy of the association rule set using AR_{AL} is maximized.

10.3.2 Modeling Tree-Structured Data

An example of three user sessions logged into an academic institution website server is depicted here to represent the process of tree-structured data representation for data mining purposes. Table 10.3 is an example of a string to integer mapping from the user sessions in Table 10.2. The mapping process from string to integer can be done with a hash function as discussed in [51]. Representing a label as an integer instead of a string label has considerable performance and space advantages [42].

As mentioned earlier, a common way of representing trees is to use the pre-order (depth-first) string encoding (φ) as described in [51]. For example, the pre-order string encoding representation of the underlying tree structure of the user navigation of Table 10.2 is transformed to (φ)(session 0) = '0 1 2 3 −1 4 −1 −1 5 6 −1 −1 −1 7 8 9 −1 −1 −1 10 11 −1 12 −1 −1' and (φ)(session 1) = '0 1 13 14 −1 15 −1 −1 16 −1 −1 17 −1' and (φ)(session 2) = '0 1 18 19 −1 −1 −1 20 21 −1 −1 7 22 −1 −1'. The access sequence of web pages from Table 10.2 can be represented in a tree-structured way as shown in Fig. 10.3. The order of pages accessed is reflected by the pre-order traversal of the tree. The corresponding tree structure is more informative than just a sequence of pages accessed as it captures the structure of the web site, and navigational patterns over this website, and the discovered knowledge patterns will as a result be more informative and useful, as already elaborated on in works such as [16, 17, 51, 52]. With this approach, specific pages can be considered within the same context. An example of this is the two pages being grouped under the 'centres and labs' parent node with label 13 in the tree of session 1, and 2 pages under the 'research' parent node with label 1 in the tree of session 0. Session 0 has come from an IP within the university and is most likely an example of a student acquiring some general information about the institute and then seeking information related to postgraduate study. The first session came from an IP internal to the university, where the user was interested in looking for jobs by browsing institutional centres and labs, and contacted the institute for more information. While session two may come from a potential external student who is searching for a potential supervisor by browsing some related conference papers and is interested in finding a research training program.

Table 10.2 Example of user session

Session 1:
/
/research.html
/research/topics.html
/research/topics/51-business-intelligence.html
/research/topics/55-e-education-ecosystems.html
/research/seminars.html
/research/seminars/413-presentation-by-eric-feinberg.html
/phd-a-msc.html
/phd-a-msc/scholarships.html
/phd-a-msc/scholarships.html#debii
/about.html
/about/objectives.html
/about/mission-and-vision.html
Session 2:
/
/research.html
/research/centres-and-labs.html
/centres-and-labs/217-anti-spam-research-lab-asrl.html
/centres-and-labs/214-centre-for-stringology-a-applications
-csa-.html
/research/jobs.html
/contact-us.html
Session3:
/
/research.html
/research/publications.html
/research/publications/conf-a-journal-papers.html
/allstaff.html
/allstaff/Research Professors & Fellows.html
/exchange-students.html
/phd-a-msc.html
/phd-a-msc/research-training.html

The integer-indexed tree is then formatted as shown in Table 10.4. This dataset format representation was proposed by [51]. Please note that the second column (*cid*) could be used to refer to a specific entity which the record describes (e.g. User id). However, in many domains such information is often unavailable, or it has been intentionally omitted or related through the transaction id (*tid*). Hence, in most of the tree databases represented in this format, the *cid* column will simple be a repetition of the *tid* column. This is the common format used in the frequent subtree mining field [17].

Table 10.3 Integer mapping for web pages from Table 10.2

ID	Web page
0	Homepage
1	Research
2	Topics
3	51-business-intelligence
4	55-e-education-ecosystems
5	Seminars
6	413-presentation-by-eric-feinberg
7	phd-a-msc
8	Scholarships
9	scholarships.html#debii
10	About
11	Objectives
12	Mission-and-vision
13	Centres-and-labs
14	217-anti-spam-research-lab-asrl
15	214-centre-for-stringoLogsy-a-applications-csa-
16	Jobs
17	Contact-us
18	Publications
19	Conf-a-journal-papers
20	Allstaff
21	Research Professors & Fellows
22	Research-training

10.3.3 Database Structure Model (DSM)

The definition given by [14] is utilized here to describe the *Database Structure Model* (DSM). Generally, the string-like representation of a tree database (example given in Table 10.4, is converted into a flat data format while preserving the ancestor-descendant and sibling node relationships. Henceforth, this structure-preserving flat data representation will be simply referred to as 'table'. The header of the table contains the DSM without any specific attribute names. It represents only the most general structure where every instance from the tree database can be matched to. This will ensure that when the labels of a particular transaction from the tree database are processed, they are placed in the correct column, corresponding to the position in the DSM that this label matches. To illustrate the complete conversion process using DSM, please refer to Fig. 10.2. Using the string encoding format representation [51], the tree database T_{db} from Fig. 10.2 would be represented as is shown in Table 10.1, where the left column corresponds to the transaction identifiers, and the right column is the string encoding of each subtree.

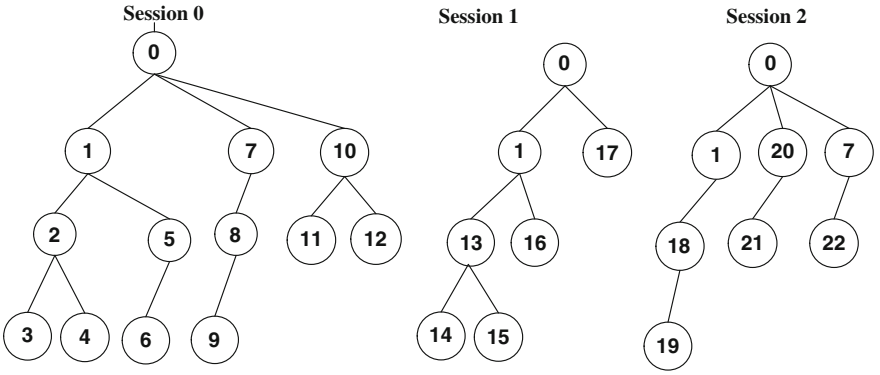

Fig. 10.3 Integer-indexed tree of user sessions in Table 10.2

Table 10.4 An integer-indexed tree in Fig.10.3 formatted as a string-like representation as used in [51]

tid	*cid*	$\|S\|$	Pre-order(depth-first) encoding
0	0	25	0 1 2 3 −1 4 −1 −1 5 6 −1 −1 −1 7 8 9 −1 −1 −1 10 11 −1 12 −1 −1
1	1	25	0 1 13 14 −1 15 −1 −1 16 −1 −1 17 −1
2	2	25	0 1 18 19 −1 −1 −1 20 21 −1 −1 7 22 −1 −1
–	–	–	–

tid transaction-id, *cid* omitted (i.e. equal to *tid*), $|S|$ size of string

In this example, the DSM is reflected in the structure of T_6 in Fig. 10.2 and it becomes the header of the table to reflect the attribute names as explained previously. The string encoding is used to represent this uniform structure and since the order of the nodes (and backtracks ('–1')) is important, the nodes and backtracks are labeled sequentially according to their occurrence in the string encoding. For nodes (labels in the string encoding), x_i is used as the attribute name, where i corresponds to the pre-order position of the node in the tree, while for backtracks, b_j is used as the attribute name, where j corresponds to the backtrack number in the string encoding. Hence, from our example in Fig. 10.2 and Table 10.1, (DSM)='$x_0, x_1, x_2, x_3, b_0, x_4, b_1, b_2, b_3, x_5, x_6, x_7, b_4, x_8, b_5, b_6, b_7$'.

To fill in the remaining rows, every transaction from T_{db} is scanned and when a label is encountered, it is placed in the matching column (i.e. under the matching node (x_i) in the DSM), and when a backtrack ('-1') is encountered, a value '1' (or 'y') is placed in the matching column (i.e. matching backtrack (b_j) in DSM). The remaining entries are assigned values of '0' (or 'No', indicating non existence). The flat data format of T_{db} from Table 10.1 (and Fig. 10.2) is illustrated in Table 10.5.

The conversion process can be formalized as follows. Let the tree database consisting of n transactions be denoted as $T_{db} = \{tid_0, tid_1, \ldots, tid_{n-1}\}$, and let the string encoding of the tree instance at transaction tid_i be denoted as $\varphi(tid_i)$. The DSM is extracted from T_{db} using the procedure explained earlier. Further, let $|\varphi(tid_i)|$ denote the number of elements in $\varphi(tid_i)$, and $\varphi(tid_i)_k$ ($k = \{0, 1, \ldots, |\varphi(tid_i)| - 1\}$) denote

Table 10.5 Flat representation of T_{db} in Fig. 10.2 and Table 10.1

x_0	x_1	x_2	x_3	b_0	x_4	b_1	b_2	b_3	x_5	x_6	x_7	b_4	x_8	b_5	b_6	b_7
a	b	d	n	1	e	1	1	1	c	0	0	0	0	0	0	1
b	c	0	0	0	0	0	0	1	b	e	0	0	0	0	1	1
b	d	e	0	0	0	0	1	1	0	0	0	0	0	0	0	0
l	m	0	0	0	0	0	0	1	n	0	0	0	0	0	0	1
k	l	m	0	0	0	0	1	1	n	0	0	0	0	0	0	1
b	a	c	f	1	0	0	1	1	d	0	0	0	0	0	0	1
a	b	c	d	1	e	1	1	1	f	g	h	1	i	1	1	1

Table 10.6 Flat representation of *Tdb* in Fig. 10.2 and Table 10.1 when minimum support = 3

x_0	x_1	x_2	x_3	b_0	b_1	b_2	x_4	b_3
a	b	c	n	1	1	1	c	1
b	c	0	0	0	0	1	b	1
b	d	e	0	0	1	1	0	0
l	m	0	0	0	0	1	n	1
k	l	m	0	0	1	1	n	1
b	a	c	f	1	1	1	d	1
a	b	c	d	1	1	1	f	1

the kth element (a label or a backtrack '-1') of $\varphi(tid_i)$. The flat data format or table F_T $(C, R)(C = \text{columns}, R = \text{rows})$ is set up where $C = \{c_0, c_1, \ldots, c_{m-1}\}(m = |C| = |\varphi(DSM)|)$, and $R = \{r_0, r_1, \ldots, r_{p-1}\}(p = |R| = n + 1)$ (i.e. extra column for attribute names). The value in column number x and row number y is denoted as $F_T(c_x, r_y)$. Hence, to set the attribute names $F_T(c_i, r_0) = \varphi(DSM)_k$ where $i = k = \{0, 1, \ldots, (|\varphi(DSM)| - 1)\}$.

In addition, during the conversion process as mentioned in [16], one can incorporate the minimum support threshold s so that the DSM captures only those structural characteristics that have occurred in at least $s\,\%$ of the tree database. Hence, in some cases only a fraction of a tree instance can be matched to the DSM due to low occurrences in the tree database, but the partial information still needs to be included in the resulting flat table. As an example, refer to the tree database T_{db} in Table 10.5 and Fig. 10.2, in mining the subtrees with minimum support threshold of 3, the resulting DSM would be as follows: 'x_0, x_1, x_2, x_3, b_0, b_1, b_2, x_4, b_3' and the new table is shown in Table 10.6.

10.3.4 Tree to Flat Conversion Example Using Academic Institution WebLogs Data

Referring to the an Academic Institution WebLogs data example in Sect. 10.3.2, the pre-order encoding format of the tree database needs to be converted into a flat representation as proposed by [14]. The DSM applications were described earlier in

Sect. 10.3.3. In this section, an illustrative example is provided using an Academic Institution WebLogs example as reference. The DSM is reflected in the structure of T_0 in Table 10.4 and the corresponding tree is shown in Fig. 10.3 (Session 0). Transaction T_0 becomes the general structure of DSM and the header in Table 10.7 to reflect the attributes names. Every transaction that remains in the T_{db} will be matched against the DSM and every node label placed in the matching column (i.e. under the matching node (x_i) in the DSM). The flat data format of T_{db} from Table 10.4 is illustrated in Table 10.7.

10.3.5 Representing Disconnected Trees w.r.t. DSM

As discussed earlier in Sect. 10.3.3, the rules from DSM can be converted into pre-order string encoding of the subtrees, and hence are represented as subtrees of the tree database. However, some rules may not be representatives of valid subtrees. For example, it is possible that some items in the rules correspond to sibling nodes in the original tree, while the parent or any ancestor node connecting those in the original tree is not present in the rules discovered using DSM approach. Hence, this would result in an invalid subtree as the nodes are disconnected. In addressing this matter, one can add the other nodes that make it into a valid subtree but flag them as irrelevant. The process consists of sequentially listing the values of each matched node in DSM, while retaining the level of embedding information of each current node in DSM and in the subtree pattern. Since the DSM itself is ordered according to the pre-order traversal, this results in pre-order string encodings of the subtrees.

As a simple illustrative example, consider the following associations/patterns extracted from an Academic Institution WebLogs Data:

P1: business-intelligence human-space-computing phd-msc,

P2: scholarships management phd-a-msc.

With respect to pattern (P_1) in Fig. 10.4 and pattern (P_2) in Fig. 10.5, the items (nodes) in the rule correspond to sibling nodes in the original tree, while the parent or any ancestor node connecting those in the original tree is not present in the rule. Hence, this would result in an invalid subtree as the nodes are disconnected. This is illustrated in both Figs. 10.4 and 10.5, where irrelevant nodes are shaded grey. One can also choose to display the labels of nodes that are there to contextualize the information, i.e. scholarships and management and phd-a-msc, which would essentially contextualize the specific rule constraints. Additionally, the labels of nodes can be displayed in order to contextualize the information in the tree. In this work, these rules are recognized as FullTree rules.

Table 10.7 Flat representation of an Academic Institution WebLogs T_{db} in Table 10.4

x_0	x_1	x_2	x_3	b_0	x_4	b_1	b_2	x_5	x_6	b_3	b_4	b_5	x_7	x_8	x_9	b_6	b_7	b_8	x_{10}	x_{11}	b_9	x_{12}	b_{10}	b_{11}
0	1	2	3	Yes	4	Yes	Yes	5	6	Yes	Yes	Yes	7	8	9	Yes	Yes	Yes	10	11	Yes	12	Yes	Yes
0	1	13	14	Yes	15	Yes	Yes	16	No	No	Yes	Yes	17	No	No	No	No	Yes	No	No	No	No	No	No
0	1	18	19	Yes	No	No	Yes	No	No	No	No	Yes	20	21	No	No	Yes	Yes	7	22	Yes	No	No	Yes
–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

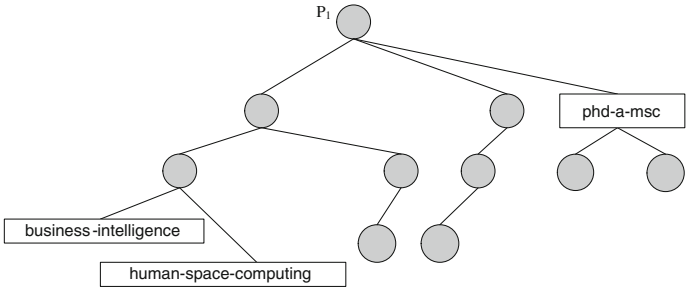

Fig. 10.4 Displaying pattern (P_1) w.r.t. DSM in Table 10.7

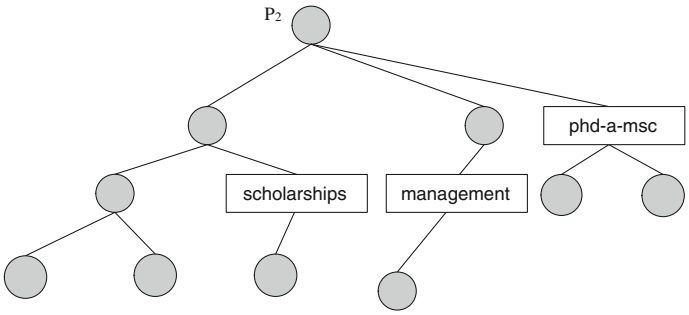

Fig. 10.5 Displaying pattern (P_2) w.r.t. DSM in Table 10.7

10.4 Method and Experimental Setup

The method used here is the integration of rule optimization framework presented in [37, 38] and structure-preserving flat representation of tree-structured data presented in [14], which as a result will allow direct application of standard statistical measures to tree-structured data. Figure 10.6 shows the proposed framework which in itself describes the experimental process. The database structure model (DSM) [14] is extracted from the tree-structured data/XML documents to preserve the structural characteristics of the data. The extracted DSM is used to create the flat representation of the tree structured data (shown in Fig. 10.6 with the square dash line region). An example of the conversion process is given in Sect. 10.3. Once the tree-structured dataset has been converted to a flat table format (FDT), the dataset is then divided into two parts. The first part is used for frequent pattern mining, statistical evaluation and rule filtering process, while the second part acts as sample data drawn from the dataset used to verify the accuracy and coverage of the discovered rules. In the pre-processing phase, missing values are handled using common distribution-based missing value imputation [27] and equal width binning approach is utilised to discretise the values of any continuous attributes. The equal-width binning approach groups the data into several buckets or bins of the same interval size. The equal width binning will be implemented based on the following steps [35]: (1) Calculate the range of variable

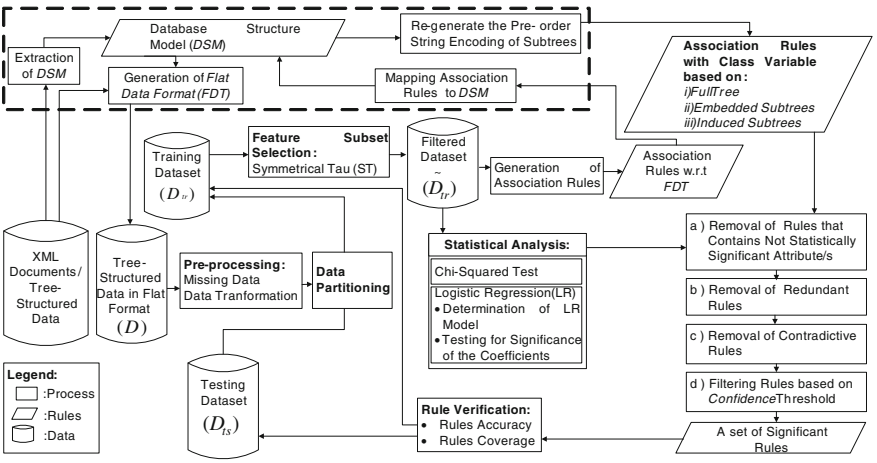

Fig. 10.6 Method and experimental setup

to be binned; (2) Using the specified number of bins, calculate the boundary (width) of each bin; (3) Using specified boundaries, assign each value of the variable to a bin for each record. The data partitioning, missing value imputation and discretization were performed using the SAS Enterprise Miner software (please refer to [35] for further detail on the use of software for data pre-processing). Secondly, feature subset selection based on attribute ranking according to Symmetrical Tau measure [54] of predictive capability is performed as described in [15].

The association rule mining algorithm is utilized to discover frequent rules from the FDT and rule filtering process based on sequence of chi-square test, Logistic Regression model selection, redundant rule removal (based on minimum improvement redundant rule constraint [4]) and optional filtering based on higher confidence threshold is performed. The extracted association rules are mapped onto the DSM (by the pre-order position of each item) to re-generate the pre-order string encoding of subtrees, thereby representing them as subtrees of the tree database.

These rules may contain both valid and invalid subtrees (disconnected subtrees), and we will refer to these as *FullTree*. In addition, the rules based on embedded subtrees and the rules based on induced subtrees (the rule sets that exclude disconnected subtrees) have also been revealed within the extracted rules. Finally the rule accuracy and coverage rate is calculated for all rule sets at different stages.The extracted frequent rules are mapped onto the DSM to re-generate the pre-order string encoding of subtrees, thereby representing them as subtrees of the tree database.

Tree-Structured Data Format Conversion: For given tree-structured data, the enumeration of all possible subtrees in a complete, non-redundant and efficient way is the major problem one needs to tackle [43]. A significant delay in the subtree patterns analysis and interpretation process may occur at lower support thresholds. Additionally, as a large number of frequent subtree patterns may be discovered, many of which may not be useful, one needs to filter out many of the irrelevant/uninteresting patterns.

The flat data format (relational or vectorial data) was proven to be acceptable and successful when utilized with many well-established data mining techniques. Thus, an effective way proposed in [14] known as Database Structure Model (DSM) is utilized in this research to represent tree-structured data in a structure-preserving flat data format. This approach offers a way of preserving tree-structured and attribute-value information. With the application of DSM, the structural characteristics are preserved during the data mining process. The extracted rules from the data mining application can be mapped onto the DSM to re-generate the pre-order string encoding of subtrees.

Let a tree structure data in flat table format (FDT) dataset be denoted as D, $I = \{i_1, i_2, \ldots, i_{|I|}\}$ the set of distinct items in D, $AT = \{at_1, at_2, \ldots, at_{|AT|}\}$ the set of input attributes in D, and $Y = \{y_1, y_2, \ldots, y_{|Y|}\}$ the class attribute with a set of class labels in D. Assume that D contains a set of n records $D = \{x_r, y_r\}_{r=1}^{n}$, where $x_r \subseteq I$ is an item or a set of items and $y_r \in Y$ is a class label, then $|x_r| = |AT|$ and $x_r = \{at_1 val_r, at_2 val_r, \ldots, at_{|AT|} val_r\}$ contains the attribute names and corresponding values for record r in D for each attribute at in AT. The training dataset is denoted as $D_{tr} \subseteq D$ and the testing dataset as $D_{ts} \subseteq D$, and filtered database after feature selection as D'_{tr} where $I' \subseteq I$.

We extracted the rule sets extracted from the flat table format (FDT) satisfying the minimum support and confidence threshold (denoted as $F(A)$). Individual rules are denoted as $fA \in F(A)$, of the form $x \rightarrow y$, where x is the antecedent and y the consequent, $\exists\{x_r, y_r\} \in D'_{tr}, x \subseteq x_r, x_r = \{at_1 val_r, at_2 val_r, \ldots, at_{|AT|} val_r\}$ and $y \in Y$ is a class label. For generating $F(A)$, SAS Enterprise Miner software was used.

Feature Subset Selection: The Symmetrical Tau (ST) measure [54] was derived from the Goodman's and Kruskal's Asymmetrical Tau measure of association for cross-classification tasks in the statistical domain. Zhou and Dillon [54] have used the Asymmetrical Tau measure as feature selection during decision tree building, and have found that it tends to favour attributes with more values. When the classes of an attribute A are increased by class subdivision, more is known about attribute A and the probability error in predicting the class of another attribute B may decrease. On the other hand, attribute A becomes more complex, potentially causing an increase in the probability error in predicting its category according to the category of B. This trade off effect inspired Zhou and Dillon [54] to combine the two asymmetrical measures in order to obtain a balanced feature selection criterion which is in turn symmetrical. However, note that in case of Boolean variables, symmetrical and asymmetrical tau will have the same value. Some powerful properties of ST, as reported in [54], are noise handling through built-in statistical strength, potential classification uncertainties are conveyed through dynamic error estimation, no bias towards multi-valued attributes, not proportional to sample size, proportional-reduction-in-error nature allows measuring of sequential variation in predictive capability, and handling of Boolean combinations of logical features.

Let there be R rows and C columns in the contingency table for attributes at_i and Y. The probability that an individual belongs to row category r and column category c is represented as $P(rc)$, and $P(r+)$ and $P(+c)$ are the marginal probabilities in row category r and column category c respectively. The measure is

based on the probabilities of one attribute value occurring together with the value of the second attribute, and for the classification task the second attribute will correspond to a special attribute in the dataset defined as class. The ST measure for the capability of input attribute at_i in predicting the class attribute Y is defined in [54] as follows.

$$Tau(at_i, Y) = \frac{\sum_{c=1}^{C}\sum_{r=1}^{R}\frac{P(rc)^2}{P(+c)} + \sum_{r=1}^{R}\sum_{c=1}^{C}\frac{P(rc)^2}{P(r+)} - \sum_{r=1}^{R}P(r+)^2 - \sum_{c=1}^{C}P(+c)^2}{2 - \sum_{r=1}^{R}P(r+)^2 - \sum_{c=1}^{C}P(+c)^2} \quad (10.1)$$

The higher values of the ST measure would indicate better discriminating criteria (features) for the class that is to be predicted in the domain. As performed in [15], the attributes are ranked according to their decreasing ST values and a relevance cut-off point is chosen at and below which all attributes are considered as irrelevant and are discarded. The relevance cut-off was selected based on the significant difference (less than half of the previous value in the ranking) between the ST values in decreasing order. This will prevent the generation of rules which would then need to be discarded when found that they were comprised of some irrelevant attributes. In accordance with [5] we have found that mutual information typically ranks attributes with more values higher than the ST measure does.

Chi-square: A natural way to express the dependence between antecedent and the consequence of an association rule is the correlation based on the chi-square test for independence [7]. The chi-square test is defined as follows: For a given D'_{tr}, the occurrence of at_i where $at_i \in AT$, $(i = (1, \ldots, |AT|)$ is independent of the occurrence of $y_r \in Y$ if $P(at_i \cup y_r) = P(at_i)P(y_r)$; otherwise at_i and y_r are dependent and correlated. The correlation between at_i and $y_r \in Y$ is measured using Eq. 10.2. For a given lift measure [40] based on Eq. 10.2, the chi-square χ^2 statistic value was utilised to determine whether the correlation is statistically significant.

$$lift(at_i, y_r) = \frac{P(at_i \cup y_r)}{P(at_1)P(y_r)} \quad (10.2)$$

Hence, the chi-square test discards any $fA_k \in F(A)$ for which $\exists at_i$ contained in x of $x \rightarrow y$, the χ^2 value is not significant for $y \in Y$ (correlation analysis in Eq. 10.2).

Logistic Regression: Another form of statistical analysis applied was the logistic regression. The relationship between the antecedent and consequent in association rule mining can be presented as a relationship between a target variable and the input variables in logistic regression. The following is the definition of the logistic regression model involved in the framework. For a given D'_{tr}, several logistic regression models were developed based on $ln(Y) = \beta_0 + \beta_1 at_1 + \beta_2 at_2 + \cdots + \beta_{|AT|} at_{|AT|} + e$, where $ln(Y)$ is the natural logarithm of the odds ratio, $\beta_0, \beta_1, \ldots, \beta_{|AT|}$ are the coefficients of the input attributes at_i, e is the error variable and Y the dichotomous class attribute. The coefficient β_i of at_i is determined based on the log likelihood value given in Eq. 10.3, where $at_i val_r$ denotes the value of attribute at_i occurring in record r.

$$\beta_i at_i = \sum_{r=1}^{n} \{y_r ln[\pi(at_i val_r)] + (1 - y_r) ln[1 - \pi(at_i val_r)]\} \tag{10.3}$$

The statistical hypothesis is then used to determine whether the input attributes are significantly related to the class attribute. A number of models can be developed from logistic regression analysis, and each produces a different selection of attributes. The model that fits the data well and has the highest predictive capability is selected. Hence, logistic regression is used to discard any $fA_k \in F(A), fB_k \in F(B), fC_k \in F(C)$ for which $\exists at_i$ contained in x of $x \rightarrow y$, the $\beta_i at_i$ value is not significant towards the class attribute Y (logistic regression analysis in Eq. 10.3).

Redundant and Contradictive Rule Removal: To remove redundant rules, we utilize the concept of productive rules [4]. This approach is based on minimum improvement redundant rule constraint [4], which discards any rule $x \rightarrow y$ if confidence $(x \rightarrow y) \leq \max(\text{confidence}(z \rightarrow y)) \forall z \subset x$. In other words, a rule $x \rightarrow y$ with confidence value $c1$ is considered as redundant if there exists another rule $z \rightarrow y$ with confidence value $c2$, where $z \subset x$ and $c1 \leq c2$. The contradictory rule constraint [53] is then utilised to discard two or more rules that have the same precedent but imply a different class value.

Rules Accuracy and Rules Coverage: A measure needs to be applied to verify whether the removal of a large volume of rules based on statistical analysis, and redundancy and contradictory assessment methods, will enable the discovery of all the interesting and significant subtree patterns. As such, the quality of the subtree pattern will be demonstrated based on their accuracy and coverage values. The values for rule accuracy and coverage will be measured at every stage and sequence of this task. This measure is crucial as it can determine the quality of the discovered rules. Additionally, this analysis will reveal the balancing/optimization issues with regards to the trade-off between accuracy rate and coverage rate.

10.5 Experimental Evaluation

In this section we present the experiments performed using the CRM dataset (real estate property management records in XML), CSLOGS dataset (web access trees) and an academic institution dataset (web access trees), structural characteristics of which are shown in Table 10.8, and the following notation is used: $|Tr|$—Number of transactions (independent tree instances); $|L|$—Number of unique labels; $|T|$—Number of nodes (size) in a transaction; $|D|$—Depth; $|F|$—Fan-out-factor (or degree). Please note, that in [52] where the structural/XML classificatotion was first proposed, it was demonstrated that a simpler classifier that does not take the structure into the account cannot achieve equally good results. Similarly, in [51] it was empirically shown, that tree-structured web-browsing patterns are more informative and useful than, their itemset/sequential pattern counter part. Hence, this study is not repeated in this work, but rather an experimental study is presented on the use of

Table 10.8 Structural characteristics of the data

	$\|Tr\|$	$\|L\|$	$Avg\|T\|$	$Avg\|D\|$	$Avg\|F\|$	$Max\|T\|$	$Max\|D\|$	$Max\|F\|$
CRM	1,181	10,611	52.97	4.89	8	533	5	46
CSLOGS	68,302	16,207	7.8	3.45	1.82	313	123	137
Academic Institution Website	18,836	34,052	9.63	4.98	1.56	60	59	37

standard statistical techniques to reduce the huge number of rules typically generated during frequent subtree mining, in the context of associative classification. As such, the focus is on the use of basic accuracy and coverage rate rule evaluation measures to observe the gradual difference in the rule set accuracy and coverage as different feature/rule filtering techniques are applied.

Each dataset underwent conversion into a structure-preserving flat data format (henceforth FDT) using the DSM approach. The backtrack attributes information was kept in DSM as this is important for preserving the structural information. Hence, this can be used to represent the resulting rules as trees/subtrees. The backtrack attributes can be optionally kept in the FDT as when present in rules, they indicate the existence/non-existence of a node irrespective of the label as discussed in [16]. We have compared the results when rules are generated from itemsets including the backtrack attributes and without, and the difference was not substantial to make it worth reporting. Inclusion of backtrack attributes typically resulted in slightly better results, in terms of increased rule set coverage rate and thus all experiments presented are done using this option. When reporting the results, the following notation will be used ST—Symmetrical Tau, AR—accuracy rate, CR—coverage rate, *FullTree*—the initial rule set containing disconnected subtree and backtrack attribute based rules, *Embedded*—after itemsets have been mapped to DSM (by pre-order positions) to generate valid connected subtrees, and *Induced*—only subtrees where maximum level of embedding is limited to 1 (i.e. parent-child relationships among the nodes, see Sect. 10.3).

10.5.1 Experiment Set 1—CRM Data

CRM data is a real-world dataset relating to the handling of complaints in the area of real estate. Each complaint relates to a particular defect in the property, and a property manager will assign a case to each defect, containing information such as case managers, contractors, areas of defect, district and building type. The classification problem considered corresponds to the "WorkCompletion", with 2 possible values (within a month and more than a month duration. The attributes containing similar information or referring to work/task completion duration have then been removed. The dataset consists of 1,181 instances with 675 attributes, of which 66 % was used

Table 10.9 Subtree association rule evaluation for CRM data

Type of analysis	Data partition	*FullTree*			*Induced*		
		# of Rules	AR %	CR %	# of Rules	AR %	CR %
# of Rules after ST	Training	27116	83.02	100	5270	81.56	100
	Testing		83.74	100		83.4	100
Logistic regression	Training	91	79.85	100	17	68.54	100
	Testing		80.95	100		70.57	100
Redundancy removal	Training	51	76.78	100	17	68.54	100
	Testing		77.72	100		70.57	100
Min. Conf. 60 %	Training	44	83.82	95.50	12	77.20	91.53
	Testing		84.57	96.15		79.18	93.59

for training and 34 % for testing. However, there are many complex classes within this CRM data which may interest the users of the data. Nevertheless in this case, as our main purposes is not to analyse the problem of CRM itself, but to look at the CRM data as an example of tree-structured data, the attention is confined to the aforementioned class. The resulting DSM based flat data format contains 675 attributes (including the class), 586 selected attributes based on Symmetrical Tau(ST) feature selection. The rules are then generated based on support of 5 % and confidence of 50 %. Note that initially the dataset with backtrack attributes was used, which caused memory issues in the SAS software and hence we applied the ST feature selection prior to generating association rules which removed all of the backtrack attributes in this dataset. Furthermore, for this dataset, all subtrees generated are of induced type, and hence we do not report any results for the embedded subbtree variation as it is identical to induced for this data.

Table 10.9 shows the results as the statistical analysis and the redundancy assessment have progressively been utilized to evaluate the interestingness of rules. Note that chi-square analysis is not presented as it did not result in any rule removal at that stage, and all of the connected subtrees were of induced subtree type in this dataset. As one can see a significant number of rules was removed by applying the logistics regression analysis, and in *FullTree* rule set further 40 rules were detected as redundant. This has reduced the AR % by about 3 %, but after rules whose minimum confidence is below 60 % have been removed (last row) the accuracy has increased with the cost of not covering around 5 % of the instances. In this experiment, *FullTree* rule set is the most optimal one, as it is not only more accurate in classifying/predicting specific instances in the database, but also achieves a higher coverage rate in the final step compared to *Induced* rule set. The *FullTree* rule set can contain rules that do not convert to valid (connected) subtrees when matched to DSM. Nevertheless, these are important to include as they may represent important associations that should not be lost because they do not convert to connected valid subtrees. Note that we have tried to run the XRules structural classifier [52] on this data, but since there are quite a few repeating node labels in single tree instances, caused by repetition of defects and individual cases within a single record, the tree

Table 10.10 CSLOGS flattened data characteristics and initial number of rules for varying support

Support threshold (%)	Atrr. # DSM flat	# Selected attr. Sym. Tau	# of Rules with target attr.		
			FullTree	*Embedded*	*Induced*
1	222	217	13835	13833	13809
5	64	52	920	919	918
10	40	29	216	215	215
20	24	11	48	47	47
30	16	7	32	31	31

mining algorithm [51] on which the XRules is based on, has difficulties in extracting subtrees at required low support thresholds.

10.5.2 Experiment Set 2—CSLOGS Data

The CSLogs data comprises the web access trees from the computer science department of the Rensselaer Polytechnic Institute previously used in [52] to evaluate the XRules structural classifier. All of the three datasets (US1924, US2430, and US304) were combined and instances were replicated (in both training and test data) to make the class distribution even. The tree instances are labelled according to two classes, namely the internal and external web site access. The total number of combined instances is 68302. The training set was comprised of 66 % of the data and the remainder was left as the test set. Since different support thresholds were used, in our approach the flat data representation of the dataset is done separately for each support threshold, as the extracted database structure model (DSM) varies; hence, the number of attributes used during frequent pattern generation. The general characteristics of the flat data format (including backtrack attributes) and initial number of rules extracted for CSLogs data (50 % minimum confidence) at varying support thresholds is provided in Table 10.10. Note, that when using the association rules for classification task it is natural that performance will vary depending on the support threshold used. Hence, different support thresholds were tried from a larger to a smaller extreme, and as expected for larger support thresholds there will be a trade-off for limited coverage as only the very frequent subtrees will be extracted to form part of the model.

For this dataset, the best results were achieved for the lowest examined support threshold of 1 %, and detailed results of progressively filtered rules based on statistical analysis and redundancy removal are presented in Table 10.11 for support 1 % (at the end of this subsection we present the performance of final rule sets for all the support thresholds). The number of rules are shown in brackets below each AR and CR values reported. The results reveal that by selecting important input attributes with ST and evaluating the rules with statistical analysis and redundancy assessment method,

Table 10.11 Subtree association rule evaluation for CSLOG data (1 % support 50 % confidence)

Type of analysis	Data partition	*FullTree*		*Embedded*		*Induced*	
		AR %	CR %	AR %	CR %	AR %	CR %
Initial rules	Training	68.09	98.59	68.12	98.59	68.11	98.59
		(13835)	(13835)	(13834)	(13834)	(13810)	(13810)
	Testing	69.94	98.6	69.94	98.6	69.94	98.6
Rules after ST	Training	69.94	98.59	70.02	98.59	70.02	98.59
		(6084)	(6084)	(6083)	(6083)	(6081)	(6081)
	Testing	72.01	98.6	72.1	98.6	72.1	98.6
Chi-Square	Training	79.22	48.97	79.02	48.39	78.41	48.39
		(73)	(73)	(72)	(72)	(65)	(65)
	Testing	78.78	48.77	78.57	48.25	78.06	48.25
Logistic regression	Training	79.22	48.97	79.02	48.39	78.41	48.39
		(73)	(73)	(71)	(71)	(64)	(64)
	Testing	78.78	48.77	78.57	48.25	78.06	48.25
Redundancy removal	Training	79.02	48.97	78.71	48.97	78.71	48.97
		(61)	(61)	(54)	(54)	(54)	(54)
	Testing	78.53	48.77	78.53	48.77	78.53	48.77

there is a significant reduction in the number of rules. While an increase in AR can be observed, this is at the cost of reduced CR capabilities. The characteristics of the *FullTree* rule set are similar to those of the *Embedded* and *Induced* rule sets, and the AR and CR are very similar or the same for the different rule sets. This is because the rules from *Embedded* and *Induced* rule sets are subsets of *FullTree*, and in this dataset there were not so many variations among the rule sets w.r.t the level of embedding in subtrees or frequent patterns that produce disconnected subtrees. To conclude, the increase in prediction/classification accuracy comes with a trade-off since fewer instances are captured from the datasets. On the positive side, a smaller number of rules is expected to have better generalization power and are easier for the user to understand and utilize for decision support purposes.

Comparison with XRules for varying support thresholds. In Table 10.12 we compare the AR and CR of the final rule sets of *FullTree* with XRules approach for varying support thresholds. Note that the approaches are fairly different in terms of the rule filtering performed in the process. Nevertheless, the comparison performed serves mainly as a benchmark for the kind of accuracy and coverage rate that is to be obtained when basing the classification on frequent patterns/subtrees extracted using the support and confidence thresholds. As such, in no way do the results indicate that one approach performs better than the other, as the internal mechanism is rather incompatible. The XRules approach is based on the TreeMiner [51] algorithm for extracting ordered embedded subtrees, and hence the number of rules extractd at varying support thresholds is larger (shown in brackets), since the likelihood that a subtree will be frequent when it does not need to occur at the same position is much

Table 10.12 Comparison of rules accuracy and coverage rate for CSLogs data using the XRules and *FullTree* final rule set

Support	1 %		5 %		10 %	
	AR %	CR %	AR %	CR %	AR %	CR %
XRules	72.72	66.04	61.74	40.7	56.9	23.21
	(298)	(298)	(20)	(20)	(3)	(3)
FullTree	78.53	48.77	78.73	20.35	76.9	20.3
	(61)	(61)	(4)	(4)	(2)	(2)

Table 10.13 Rule sets at support 10 %

#	XRules	#	*FullTree*
1	1 → Class(0)	1	X1(1) → Class(0)
2	12811 → Class(1)	2	X1(12811) → Class(1)
3	6 → Class(0)		

higher. On this note, the rule sets of the XRules approach will typically have higher coverage rate, especially in the CSLOGS dataset, where subtrees do in fact occur at many different positions due to variations in website navigation. However, one can see that this is at times at a cost of reduction in AR, and constraining the subtrees by position could be seen as more precise, but naturally would cover less cases. To give a simple example, please refer to Table 10.13 where we show rule sets for the support value of 10 %. One can observe that the *FullTree* rule set does not contain a rule that corresponds to rule number 3 in XRules even though it was considered frequent by XRules. The reason for this is that the particular node with label "6" with "Class(0)", where "6" occurs at the same node/position in DSM did not occur in 10 % of the instances to be considered frequent and part of the *FullTree* rule set. The two matching rules correspond to the first page accessed during the site navigation session, as it is labelled with pre-order position 1, namely X1 in our approach (note that X0 is a virtual node in the CSLOGS dataset always labelled with 0 and is removed in both approaches). For support threshold of 20 and 30 % no rules were extracted in our approach, while XRules only had the single default rule for majority class.

10.5.3 Experiment Set 3—Academic Institution Web Log Data

Academic Institution WebLogs data is an apache2 (v2.2.3) web server logs files. The WebLogs data was initially used in [16] in utilizing the DSM application. For the purpose of the work in this research, the similar setting of the WebLogs data as described in [16] has been utilized. The data was collected for a four-month period in its native (default) format. During this period, all access to the website was stored in logs files, while messages stored in the normal error message logs were excluded. The access to the website was then classified as "internal" (within the university) and "external" (outside the university). The grouped user sessions were converted

Table 10.14 Academic Institution flattened data characteristics and initial number of rules for varying support

Support threshold (%)	Atrr. # DSM flat	# Selected attr. Sym. Tau	# of Rules with target attr.		
			FullTree	*Embedded*	*Induced*
1	442	217	–	–	–
5	126	123	28282	28282	28282
10	70	63	234	234	234
20	36	29	50	49	49
30	26	19	14	13	13

to trees as was explained with the illustrative example in Sect. 3.1. The resulting dataset had 18,836 instances, of which 66 % was used for training and the remainder for testing. The details of the setting of the WebLogs access can be found in [16]. The general characteristics of the flat data format (including backtrack attributes) and initial number of rules extracted for education institution data (50 % minimum confidence) at varying support thresholds is provided in Table 10.14.

In this dataset, similar to the experiments described in Sect. 10.5.2, rules from *FullTree*, Embedded and Induced rule sets have been progressively assessed with statistical analysis and redundancy assessment method. The results demonstrate that the conversion of the original tree-structured data into the flat data format representation, created a very large number of input attributes, especially at lower support thresholds. By utilizing the Apriori algorithm to generate all frequent rules, one might encounter difficulties in analyzing all rules given certain support and confidence constraints.

By referring to the Table 10.15, even with the given support constraint, the number of extracted rules (Initial Rule Set) is large. A large volume of rules may be discovered due to the presence of irrelevant attributes in the dataset. The capabilities of ST in selecting appropriate attributes, thereby removing irrelevant attributes, are shown in our previous experiments for relational data problems. For this particular task of evaluating tree-structured rules, similar experiments were conducted. The attributes for each different support were ranked according to their decreasing ST and a relevance cut-off point was chosen.

Table 10.15 indicates the differences between the number of initial input attributes and the number of attributes after applying Symmetrical Tau (ST) with their respective rule number (below) for each dataset for each different support. All attributes that have been removed from the WebLogs data are backtrack attributes. This indicates that the inclusion of these backtrack nodes may not be useful or have low capabilities in predicting the class attributes in this dataset.The input variable that contains a single value is unable to distinguish the class variables. Such input attributes have been discarded as they are considered irrelevant based on the ST value calculated. With the application of ST feature selection technique, rules that contain attributes that failed the ST measure are discarded. The large number of rules were managed to be reduced

Table 10.15 Subtree association rule evaluation for Academic Institution data (10 % support 50 % confidence)

Type of analysis	Data partition	*FullTree*		*Embedded*		*Induced*	
		AR %	CR %	AR %	CR %	AR %	CR %
Initial Rules	Training	64.27 (232)	100.00 (232)	64.54 (232)	100.00 (232)	64.54 (232)	100.00 (232)
	Testing	70.06 (232)	100.00 (232)	70.55 (232)	100.00 (232)	64.54 (232)	100.00 (232)
Rules after ST	Training	75.19 (43)	73.95 (43)	74.94 (43)	73.95 (43)	74.94 (43))	73.95 (43)
	Testing	74.94 (43)	74.09 (43)	74.84 (43)	74.09 (43)	74.84 (43))	74.09 (43)
Chi-Square	Training	78.21 (11)	64.47 (11)	77.56 (10)	64.47 (10)	77.56 (10)	64.47 (10)
	Testing	74.96 (11)	60.12 (11)	74.58 (10)	61.02 (10)	74.58 (10)	61.02 (10)

with a proper sequence of usage of parameters including the ST feature selection, statistical analysis and the redundancy assessment method. According to Table 10.15, with the reduction of number of rules for *FullTree*, Embedded and Induced rule sets for Academic Institution Weblogs (10 % Support) the AR are increased but at the cost of a decrease in CR. One can also notice that the AR for the *FullTree* rule set is initially slightly lower than the AR of the Embedded and Induced rule set, but after Symmetrical Tau is applied, the accuracy of *FullTree* is higher and remains higher after chi-square rule filtering. Note that for this data there were no further rules removed via logistic regression and redundancy check, and hence these stages are not shown in Table 10.15.

10.6 Conclusion and Future Work

The work presented in this chapter has explored the application of a number of statistical methods to optimize the subtree based associative classification for tree-structured data. It has utilized a structure-preserving flat format representation, to progressively apply a number of statistical methods to first filter out irrelevant attributes followed by the removal of irrelevant and redundant rules. The use of this method has implications that the subtree based association rules are restricted to those that occur at the same position in the original tree database, and that the initial rule (before subtree reconstruction), can contain rules based on disconnected subtrees. Experiments were performed on three real datasets, and using the proposed approach a large number of rules were removed in both cases without negatively affecting the accuracy of the rule set, while for more structurally varied data, this

optimization was at the cost of a large reduction in coverage rate. The results on this data were compared with a structural classifier based on traditional subtrees, and some important differences and implications were highlighted. The results show that associations based on disconnected subtrees can be useful, while the positional constraint can often result in more precise rules for structurally varied data, but at the cost of lower coverage rate. From these findings one can conclude that when forming association rules for tree-structured data, one should not be constrained to a valid and connected subtree because an interesting association can be anywhere in a tree instance, and it does not need to be a connected subtree of that instance. These findings indicate that including disconnected subtrees and constraining the subtrees by their exact occurrence in the database in addition to traditional subtree patterns, could improve the classifiers for tree-structured data. The method used in this chapter is to be seen as complementary and in no way a replacement of the traditional way that subtrees are mined.

Our future work, will investigate the application domains where including such association rules can be beneficial and the right way to combine them with traditional subtree patterns for optimal performance.

Furthermore, the chi-square and the logistic regression measures were used as a case in point for statistic-based rule filtering, while Symmetrical Tau was utilized in the feature subset selection process. However, by no means is any claim being made that these are the most optimal measures to be used for their specific purpose. In fact, we have used the confidence constraint here because of the stronger focus on statistical quality assessment and the difference between the rule sets discovered using the traditional support and confidence framework. However, many other measures could be used and applied instead of the support and or confidence constraint, which, as discussed in several works [12, 23, 29], will yield more interesting rules. Therefore, another future work will evaluate the combinations of other constraints, statistical measures and techniques for rule removal/attribute relevance determination, in context of the tree-structured data domain.

References

1. Agrawal, R., Imieliski, T., Swami, A.: Mining association rules between sets of items in large databases. ACM SIGMOD Rec. **22**(2), 207–216 (1993)
2. Aumann, Y., Lindell, Y.: A statistical theory for quantitative association rules. Intell. Inf. Syst. **20**(3), 253–283 (2003)
3. Bathoorn, R., Koopman, A., Siebes, A.: Reducing the frequent pattern set. In: Proceedings of the 6th IEEE International Conference on Data Mining—Workshops, pp. 55–59 (2006)
4. Bayardo, R., Agrawal, R., Gunopulos, D.: Constraint-based rule mining in large, dense databases. Data Min. Knowl. Discov. **4**(2–3), 217–240 (2000)
5. Blanchard, J., Guillet, F., Gras, R., Briand, H.: Using information-theoretic measures to assess association rule interestingness. In: Proceedings of the 5th IEEE International Conference on Data Mining, pp. 215–238 (2005)
6. Bolon-Canedo, V., Sanchez-Marono, N., Alonso-Betanzos, A.: A review of feature selection methods on synthetic data. Knowl. Inf. Syst. **34**(3), 483–519 (2013)
7. Brijs, T., Vanhoof, K., Wets, G.: Defining interestingness for association rules. Int. J. Inf. Theor. Appl. **10**(4), 370–376 (2003)

8. Brin, S., Motwani, R., Silverstein, C.: Beyond market baskets: generalizing association rules to correlations. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 265–276 (1997)
9. Cheng, H., Yan, X., Han, J., Hsu, C.W.: Discriminative frequent pattern analysis for effective classification. In: Proceedings of the 23rd International IEEE Conference on Data Engineering, pp. 716–725 (2007)
10. Cheng, H., Yan, X., Han, J., Yu, P.: Direct discriminative pattern mining for effective classification. In: Proceedings of the 24th International Conference on Data Engineering, pp. 167–178 (2008)
11. Dash, M., Liu, H.: Feature selection for classification. Intell. Data Anal. **1**(3), 131–156 (1997)
12. Geng, L., Hamilton, H.: Interestingness measures for data mining: a survey. ACM Comput. Surv. **338**(3, Article No. 9) (2006)
13. Goodman, A., Kamath, C., Kumar, V.: Data analysis in the 21st century. Stat. Anal. Data Min. **1**(1), 1–3 (2008)
14. Hadzic, F.: A structure preserving flat data format representation for tree-structured data. In: Proceedings of PAKDD Workshops, vol. 2011, pp. 221–233 (2012)
15. Hadzic, F., Dillon, T.: Using the symmetrical tau (τ) criterion for feature selection in decision tree and neural network learning. In: Proceedings of the 2nd SIAM Workshop on Feature Selection for Data Mining: Interfacing Machine Learning and Statistics (2006)
16. Hadzic, F., Hecker, M.: Alternative approach to tree-structured web log representation and mining. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 235–242 (2011)
17. Hadzic, F., Tan, H., Dillon, T.S.: Mining of Data With Complex Structures, 1st edn, Studies in Computational Intelligence, vol. 333, . Springer (2011)
18. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann (2001)
19. Hashimoto, K., Takigawa, I., Shiga, M., Kanehisa, M., Mamitsuka, H.: Mining significant tree patterns in carbohydrate sugar chains. Bioinformatics **24**(16), 167–173 (2008)
20. Knijf, J.D., Feelders, A.J.: Monotone constraints in frequent tree mining. In: Proceedings of the 14th Annual Machine Learning Conference of Belgium and the Netherlands, BENELEARN pp. 13–20 (2005)
21. Kudo, M., Sklansky, J.: Comparison of algorithms that select features for pattern classifiers. Pattern Recognit. **33**(1), 25–41 (2000)
22. Lallich, S., Teytaud, O., Prudhomme, E.: Association rule interestingness: measure and statistical validation. In: Quality Measures in Data Mining. Studies in Computational Intelligence, vol. 43, pp. 251–275. Springer (2007)
23. Lallich, S., Teytaud, O., Prudhomme, E.: Formal framework for the study of algorithmic properties of objective interestingness measures. In: Data Mining: Foundations and Intelligent Paradigms, vol. 24, pp. 77–98. ISRL (2012)
24. Le Bras, Y., Lenca, P., Lallich, S.: Mining classification rules without support: an anti-monotone property of Jaccard measure. In: Proceedings of the 14th International Conference on Discovery Science, pp. 179–193 (2011)
25. Lenca, P., Meyer, P., Vaillant, B., Lallich, S.: On selecting interestingness measures for association rules: user oriented description and multiple criteria decision aid. Eur. J. Oper. Res. **184**(2), 610–626 (2008)
26. Li, J., Shen, H., Topor, R.: Mining the optimal class association rule set. Knowl.-Based Syst. **15**(7), 399–405 (2002)
27. Little, R., Rubin, D.: Statistical Analysis with Missing Data, 2nd edn. Wiley, New York (2002)
28. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, pp. 80–86 (1998)
29. McGarry, K.: A survey of interestingness measures for knowledge discovery. Knowl. Eng. Rev. **20**(1), 39–61 (2005)
30. Molina, L., Belanche, L., Nebot, A.: Feature selection algorithms: a survey and experimental evaluation. In: Proceedings of IEEE International Conference on Data Mining, pp. 306–313 (2002)

31. Nakamura, A., Kudo, M.: Mining frequent trees with node-inclusion constraints. In: Advances in Knowledge Discovery and Data Mining, vol. 3518, pp. 850–860. Springer (2005)
32. Ozaki, T., Ohkawa, T.: New frontiers in applied data mining, PAKDD 2008 International Workshops. Mining Mutually Dependent Ordered Subtrees in Tree Databases, pp. 75–86. Springer, Heidelberg (2009)
33. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
34. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufman (1993)
35. Refaat, M.: Data Preparation for Data Mining Using SAS. Morgan Kaufmann Publishers, San Francisco (2007)
36. Roiger, R., Geatz, M.: Data Mining: A Tutorial-Based Primer. Addison Wesley, Boston (2003)
37. Shaharanee, I., Hadzic, F.: Evaluation and optimization of frequent, closed and maximal association rule based classification. Stat. Comput. **23**, 1–23 (2013)
38. Shaharanee, I., Hadzic, F., Dillon, T.: Interestingness measures for association rules based on statistical validity. Knowl.-Based Syst. **24**(3), 386–392 (2011)
39. Siebes, A., Vreeken, J., Leeuwen, M.V.: Item sets that compress. In: Proceedings of the SIAM Conference on Data Mining, pp. 393–404 (2006)
40. Silverstein, C., Brin, S., Motwani, R.: Beyond market baskets: generalizing association rules to dependence rules. Data Min. Knowl. Disc. **2**(1), 39–68 (1998)
41. Srikant, R., Vu, Q., Agrawal, R.: Mining association rules with item constraints. In: Proceedings of the 3rd Internationall Conference on Knowledge Discovery in Databases and Data Mining, pp. 67–73 (1997)
42. Tan, H., Dillon, T., Hadzic, F., Feng, L., Chang, E.: IMB3-Miner: Mining induced/embedded subtrees by constraining the level of embedding. In: Proceedings of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 450–461 (2006)
43. Tan, H., Hadzic, F., Dillon, T., Chang, E., Feng, L.: Tree model guided candidate generation for mining frequent subtrees from XML documents. ACM Trans. Knowl. Disc. Data Min. **2**(2), 1–43 (2008)
44. Tan, P., Kumar, V., Srivastava, J.: Selecting the right interestingness measure for association patterns. In: Proceedings of the 8th ACM Knowledge Discovery and Data Mining Conference, pp. 32–41 (2002)
45. Veloso, A., Meira, W., Zaki, M.: Lazy Associative classification. In: Proceedings of the 6th IEEE International Conference on Data Mining, pp. 645–654 (2006)
46. Webb, G.: Discovering significant patterns. Mach. Learn. **68**(1), 1–33 (2007)
47. Xiong, H., Tan, P.N., Kumar, V.: Hyperclique pattern discovery. Data Min. Knowl. Disc. **13**(2), 219–242 (2006)
48. Yan, X., Cheng, H., Han, J., Yu, P.S.: Mining significant graph patterns by leap search. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 433–444 (2008)
49. Yan, X., Han, J., Hsu, C.W.: Discrimantive frequent pattern analysis for effective classification. In: Proceedings of the 23rd IEEE International Conference on Data Engineering, pp. 716–725 (2007)
50. Yin, X., Han, J.: CPAR: Classification based on predictive association rules. In: Proceedings of the SIAM International Conference on Data Mining, pp. 396–376 (2003)
51. Zaki, M.: Efficiently mining frequent trees in a forest: algorithms and applications. IEEE Trans. Knowl. Data Eng. **17**(8), 1021–1035 (2005)
52. Zaki, M.J., Aggarwal, C.: XRules: an effective structural classifier for XML data. In: Proceedings of the 9th ACM Knowledge Discovery and Data Mining Conference, pp. 316–325 (2003)
53. Zhang, C., Zhang, S.: Collecting quality data for database mining. In: AI 2001: Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 2256, pp. 593–604. Springer (2001)
54. Zhou, X., Dillon, T.: A statistical-heuristic feature selection criterion for decision tree induction. IEEE Trans. Pattern Anal. Mach. Intell. **13**(8), 834–841 (1991)