To appear in Knowledge and Information Systems (pre-publication version)

StatApriori: an efficient algorithm for searching statistically significant association rules

Wilhelmiina Hämäläinen

Department of Computer Science, University of Helsinki, Finland

Abstract. Searching statistically significant association rules is an important but neglected problem. Traditional association rules do not capture the idea of statistical dependence and the resulting rules can be spurious, while the most significant rules may be missing. This leads to erroneous models and predictions which often become expensive.

The problem is computationally very difficult, because the significance is not a monotonic property. However, in this paper we prove several other properties, which can be used for pruning the search space. The properties are implemented in the StatApriori algorithm, which searches statistically significant, non-redundant association rules. Empirical experiments have shown that StatApriori is very efficient, but in the same time it finds good quality rules.

Keywords: Association rule; Statistical significance; Dependence; StatApriori; Search algorithm;

1. Introduction

Traditional *association rules* (Agrawal, Imielinski and Swami, 1993) are rules of form "if event $X = \overline{x}$ occurs, then also event $A = a$ is likely to occur". The commonness of the rule is measured by frequency $P(X = \overline{x}, A = a)$ and the strength of the rule by confidence $P(A = a|X = \overline{x})$. Usually it is required that both frequency and confidence should exceed some user-defined thresholds. The actual interestingness of the rule is usually decided afterwards, by some interestingness measure.

Often the associations are interpreted as correlations or dependencies between

Received Jan 16, 2009
Revised Apr 21, 2009
Accepted May 09, 2009

certain attribute value combinations. However, traditional association rules do not necessarily capture statistical dependencies, but they can associate absolutely independent events while ignoring strong dependencies. As a solution, it is often suggested (following the axioms by Piatetsky-Shapiro (1991)) to measure the lift (interest) instead of the confidence (e.g. (Tan, Kumar and Srivastava, 2004)). This produces also statistically more sound results, but still it is possible to find spurious rules while missing statistically significant rules. In the worst case, all discovered rules can be spurious (Webb, 2006; Webb, 2007). In practice, this means that the future data does not exhibit the discovered dependencies and the conclusions based on them are erroneous. The results can be expensive or even fatal, as the following example demonstrates.

Example 1. A biological database contains observation reports from different kinds of biotopes, like grove, marsh, waterside, coniferous forest, etc. For association analysis, each report is represented as a binary vector, listing the observed species, along with biotope characteristics. Local forestry societies as well as individual land owners can use the data when they decide e.g. fellings or protected sites.

The forestry society *FallAll* is going to drain swamps for new forests. Before any decisions are made, they search associations from the 1000 observations on marsh sides. They use minimum frequency 0.05 and minimum confidence 0.80. One discovered rule is *leather leaf* → *cloudberry* with frequency 0.06 and confidence 0.80. Since cloudberries are commercially important product, the forestry society decides to protect a marsh growing leather leaves, when other swamps are drained.

The decision is excellent for the leather leaf, but all cloudberries in the area disappear. The reason is that cloudberries require a wet swamp, while leather leaves can grow in both moist and wet sides. The only protected swamp in the area was too dry for cloudberries. This catastrophe was due to a spurious rule *leather leaf* → *cloudberry.* The rule has p-value 0.13 which means that there is 13% probability that the observed association is just due to chance.

In the same time, the forest society misses an important rule, namely *wet swamp,leather leaf* → *cloudberry.* This rule was not found, because it had too low frequency, 0.04. However, it is a strong rule with confidence 1.0. The p-value is 0.011 which indicates that the rule is quite reliable. Roughly speaking, it means that there is only 1.1% probability that the rule is spurious.

The problems of association rules and especially the frequency-confidence-framework are well-known (Webb, 2006; Webb, 2007; Berzal, Blanco, Sánchez and Miranda, 2001; Morishita and Sese, 2000), but still there have been only few attempts to solve the problem. Quite likely, the reason is purely practical: the problem has been considered computationally intractable. Statistical significance is not a monotonic property and therefore it cannot be used for pruning the search space in the same manner as the frequency.

However, when we search directly statistically significant rules (instead of sets), we can utilize other properties for efficient pruning. More efficiency is achieved by searching only non-redundant rules. Such rules are at least as good as the pruned rules, but simpler, and no information is lost. In practice, the simpler rules avoid overfitting and hold better in the future data.

In this paper, we introduce a set of properties which can be used for searching non-redundant, statistically most significant association rules. The properties are implemented in the *StatApriori* algorithm. Some of the pruning principles as

Table 1. Basic notations.

Notation	**Meaning**
$A, B, C, \ldots$	binary attributes
$a, b, c, \ldots \in \{0,1\}$	attribute values
$R = \{A_1, \ldots, A_k\}$	set of all attributes
$\lvert R\rvert = k$	number of attributes
$Dom(R) = \{0,1\}^k$	attribute space
$X, Y, Z \subseteq R$	attribute sets
$Dom(X) = \{0,1\}^l$	domain of X, $\lvert X\rvert = l$
$(X = \overline{x}) = \{(A_1 = a_1), \ldots, (A_l = a_l)\}$	event, $\lvert X\rvert = l$
$t = \{A_1 = t(A_1), \ldots, A_k = t(A_k)\}$	row (tuple, transaction)
$r = \{t_1, \ldots, t_n \mid t_i \in Dom(R)\}$	relation (data set)
$\lvert r\rvert = n$	size of relation r
$\sigma_{X=\overline{x}}(r) = \{t \in r \mid t[X] = \overline{x}\}$	set of rows where $X = \overline{x}$
$m(X = \overline{x}) = \lvert\sigma_{X=\overline{x}}(r)\rvert$	absolute frequency, number of rows where $X = \overline{x}$
$P(X = \overline{x}) = \frac{m(X=\overline{x})}{n}$	relative frequency of $X = \overline{x}$
$\gamma(X = \overline{x}, A = a) = \frac{P(X=\overline{x}, A=a)}{P(X=\overline{x})P(A=a)}$	lift of rule $X = \overline{x} \rightarrow A = a$
$i(fr, \gamma)$	measure function on frequency fr and lift γ
$I(X = \overline{x} \rightarrow A = a) = i(P(X = \overline{x}, A = a), \gamma(X = \overline{x}, A = a))$	value of the measure function for a rule
$upperbound(f)$	an upperbound for function f
$bestrule(X) = \arg\max_{A \in X}\{I(X \setminus A \rightarrow A)\}$	the best rule which can be constructed from X
$PS(X)$	property “potentially significant”; whether significant rules can be derived from X or its supersets
$minattr(X) = \arg\min\{P(A_i) \lvert A_i \in X\}$	“minimum attribute” of X; one with the lowest frequency

well as the first version of StatApriori algorithm were introduced in the ICDM conference version of this paper (Hämäläinen and Nykänen, 2008).

StatApriori does not need any minimum frequency thresholds even with large data sets, which guarantees that we can find all globally optimal, non-redundant rules with the selected significance measure. The measure function can be freely selected, as long as it is a monotonically increasing or decreasing function of frequency and lift. Due to efficient pruning and redundancy reduction, StatApriori can tackle problems which are impossible to compute with the classical Apriori.

The rest of the paper is organized as follows: the basic definitions are introduced in Section 2; assessing the statistical significance of association rules is discussed in Section 3; important pruning properties and theoretical principles are defined in Section 4; the StatApriori algorithm is described in Section 5; experiments are reported in Section 6; related research is discussed in Section 7; and the final conclusions are drawn in Section 8.

2. Basic definitions

In the following we give basic definitions of the association rule, statistical dependence, statistical significance, and redundancy. The notations are introduced in Table 1.

2.1. Association rules

Traditionally, association rules are defined in the frequency-confidence framework:

Definition 1 (Association rule). Let R be a set of binary attributes and r a relation according to R. Let $X \subsetneq R$, $A \in R \setminus X$, $\overline{x} \in Dom(X)$, and $a \in Dom(A)$.

The *confidence* of rule $(X = \overline{x}) \rightarrow (A = a)$ is

$$cf(X = \overline{x} \rightarrow A = a) = \frac{P(X = \overline{x}, A = a)}{P(X = \overline{x})} = P(A = a | X = \overline{x})$$

and the *frequency* (support) of the rule is

$$fr(X = \overline{x} \rightarrow A = a) = P(X = \overline{x}, A = a).$$

Given user-defined thresholds $min_{cf}, min_{fr} \in [0, 1]$, rule $(X = \overline{x}) \rightarrow (A = a)$ is an *association rule* in r, if

(i) $cf(X = \overline{x} \rightarrow A = a) \geq min_{cf}$, and
(ii) $fr(X = \overline{x} \rightarrow A = a) \geq min_{fr}$.

The first condition requires that an association rule should be strong enough and the second condition requires that it should be common enough. In this paper, we call rules association rules, even if no thresholds min_{fr} and min_{cf} are specified.

Usually, it is assumed that the rule contains only positive attribute values ($A_i = 1$). Now, the rule can be expressed simply by listing the attributes, e.g. $A_1, A_3, A_5 \rightarrow A_2$. Another common restriction is to assume that the consequent contains just one attribute. Often the rules with a set of attributes in the consequence can be derived afterwards. Derivation rules are simple, when the confidence is 1, but the exact calculi for different measure functions should be researched.

2.2. Statistical dependence

Statistical dependence is usually defined through statistical independence (e.g. (Silverstein, Brin and Motwani, 1998; Meo, 2000)):

Definition 2 (Independence and dependence). Let $X \subsetneq R$ and $A \in R \setminus X$ be sets of binary attributes.

Events $X = \overline{x}$ and $A = a$, $\overline{x} \in Dom(X)$, $a \in Dom(A)$, are *mutually independent*, if $P(X = \overline{x}, A = a) = P(X = \overline{x})P(A = a)$.

If the events are not independent, they are *dependent*.

The strength of the statistical dependence between $(X = \overline{x})$ and $(A = a)$ can be measured by *lift* or *interest*:

$$\gamma(X = \overline{x}, A = a) = \frac{P(X = \overline{x}, A = a)}{P(X = \overline{x})P(A = a)} = \frac{P(A = a | X = \overline{x}}{P(A = a)}.$$

In the following, we will concentrate on the dependencies between events containing only positive attributes. The lift of rule $X \rightarrow A$ is denoted simply $\gamma(X, A)$.

Lift $\gamma(X, A) > 1$ indicates positive dependency, lift $\gamma(X, A) < 1$ negative dependency, and $\gamma(X, A) = 1$ independence. We note that the above equation is not defined, when $P(A) = 0$. In this case, we define $\gamma(X, A) = 1$, because now $P(XA) = 0 = P(X)P(A)$, as required for independence.

2.3. Redundancy

A common goal in association rule discovery is to find *minimal* (or most general) *interesting rules*, and prune out *redundant rules* (Bastide, Pasquier, Taouil, Stumme and Lakhal, 2000). The reasons are twofold: First, the number of discovered rules is typically too large (even hundreds of thousands of rules) for any human interpreter. According to the *Occam's Razor principle*, it is only sensible to prune out complex rules $X \rightarrow A$, if their generalizations $Z \rightarrow A$, $Z \subsetneq X$ are at least equally interesting. The user just has to define the interestingness measure carefully, according to the modelling purposes. Second, pruning redundant rules can save the search time enormously, if it is done on-line. This is not possible with many interestingness functions, and usually the pruning is done afterwards.

In our case, the interestingness measure is the statistical significance, but in general, redundancy and minimality can be defined with respect to any other measure function.

Definition 3 (Redundant rules). Given an increasing interestingness measure I, rule $X \rightarrow A$ is redundant, if there exists rule $X' \rightarrow A'$ such that $X' \cup \{A'\} \subsetneq X \cup \{A\}$ and $I(X' \rightarrow A') \geq I(X \rightarrow A)$. If the rule is not redundant, then it is called non-redundant.

I.e. a rule is non-redundant, if all its generalizations (”parent rules”) are less significant. It is still possible that some or all of its specializations (”children rules”) are better. In the latter case, the rule is unlikely interesting itself.

Non-redundant rules can be further classified as minimal or non-minimal:

Definition 4 (Minimal rules). Non-redundant rule $X' \rightarrow A'$ is minimal, if for all rules $X \rightarrow A$, such that $X' \cup \{A'\} \subsetneq X \cup \{A\}$, $I(X \rightarrow A) \leq I(X' \rightarrow A')$.

I.e. a minimal rule is more significant than any of its parent or children rules. In the algorithmic level this means that we stop the search without checking any children rules, if we have just ensured that the rule is minimal.

We note that in the previous research word ”redundancy” has been used in two different senses. A rule can be redundant, if it contains redundant attributes (e.g. (Aggarwal and Yu, 1998)) or – in the context of frequent sets and rules – if its frequency can be derived from other frequent sets (e.g. (Liu, Li and Wong, 2008)).

3. Statistical significance

Next we consider the problem of assessing the statistical significance of an association rule. We discuss the main approaches and their problems and derive statistical measure functions for association rules.

3.1. Main approaches

The main idea of statistical significance is to estimate the probability that the observed discovery has occurred by chance. If the probability is very low, we can assume that the discovery is genuine. Otherwise, it is considered spurious and discarded. The probability can be estimated either analytically or empirically. The analytical approach is used in the traditional significance testing, while randomization tests estimate the probability empirically.

Traditional significance testing can be further divided into two main classes: a frequentist and Bayesian approach. The frequentist approach is the most commonly used and best studied (see e.g. (Freedman, Pisani and Purves, 2007, Ch. 26) or (Lindgren, 1993, Ch. 10.1)). The main idea is to estimate the probability of an observed or a rarer phenomenon under some null hypothesis. When the objective is to test the significance of the dependency between events X and A, the null hypothesis H_0 is the independence assumption: $P(X, A) = P(X)P(A)$. The task is to calculate the probability $p = P(M \geq m(X, A) \mid H_0)$ that XA occurs at least $m(X, A)$ times in the data set, if they were actually independent.

In the hypothesis testing, the p-value is compared to some pre-defined threshold α. If $p \leq \alpha$, the null hypothesis is rejected and the discovery is called significant at level α. Parameter α defines the probability of committing *type I error*, i.e. accepting a spurious rule. Another parameter, β, is used to define the probability of committing *type II error*, i.e. rejecting a true rule as non-significant. The problem is how to decide suitable thresholds, and often only the p-values are reported.

Deciding threshold α is even harder in the data mining where numerous patterns are tested. For example, if we use threshold $\alpha = 0.05$, then there is a 5% chance that a spurious rule passes the significance test. If we test 10 000 rules, it is likely that we will find 500 spurious rules. This so called *multiple testing problem* is inherent in the knowledge discovery, where we often perform an exhaustive search over all possible patterns.

As a solution, the more patterns we test, the stricter bounds for the significance we should use. The most famous correction method is *Bonferroni adjustment* (Shaffer, 1995), where the desired significance level p is divided by the number of tests m. In the association rule discovery, we can give an upper bound for the number of rules to be tested (Webb, 2007). An alternative is to control the expected number of errors among selected rules (Benjamini and Hochberg, 1995). In this paper, we do not try to solve this problem, but instead we will use the p values and corresponding measures only for ranking.

The idea of Bayesian significance testing is quite similar to the frequentist approach, but now we assign some prior probabilities $P(H_0)$ and $P(H_1)$ to null hypothesis H_0 and research hypothesis H_1. The conditional probabilities $P(M \geq m(X, A) \mid H_0)$ and $P(M \geq m(X, A) \mid H_1)$ are estimated from the data, and the posterior probabilities of hypotheses $(P(H_0 \mid M \geq m(X, A))$ and $P(H_1 \mid M \geq m(X, A)))$ are calculated by the Bayes rule. The results are asymptotically similar (under some assumptions even identical) to the traditional hypothesis testing, although the Bayesian testing is sensitive to the selected prior probabilities (Agresti and Min, 2005).

Gionis et al. (2007) have described how to use randomization tests to assess the significance of data mining results, like frequent sets. The main idea is to generate random data sets which have the same row and column marginals as the original data. The empirical p-value is defined as a proportion of random sets where the observed or more extreme phenomenon occurs. The only requirement is that it is meaningful to maintain the row and column sums. For example, in the market basket data the items are similar and their values can be swapped, but it is not sensible to swap values of different attributes like age, occupation, IQ, and gender.

3.2. Estimating p

The significance of the observed frequency $m(X, A)$ can be estimated exactly by the binomial distribution. Each row in relation r, $|r| = n$, corresponds to an independent Bernoulli trial, whose outcome is either 1 (XA occurs) or 0 (XA does not occur). All rows are mutually independent.

Assuming the independence of attributes X and A, combination XA occurs on a row with probability $P(X)P(A)$. Now the number of rows containing XA is a binomial random variable M with parameters $P(X)P(A)$ and n. The mean of M is $\mu_M = nP(X)P(A)$ and its variance is $\sigma_M^2 = nP(X)P(A)(1 - P(X)P(A))$. Probability $P(M \geq m(X, A))$ gives the significance p:

$$p = \sum_{i=m(X,A)}^{n} \binom{n}{i} (P(X)P(A))^i (1 - P(X)P(A))^{n-i}. \tag{1}$$

In this approach we make no assumptions on $P(A)$ and $P(X)$. An alternative is to assume that actual $P(X)$ and $P(A)$ are the same as observed and estimate the p-value in set $\sigma_X(r)$, where X is true. In this set, A occurs on each row with probability $P(A)$ and the probability of observing A at least $m(XA)$ times from $m(X)$ (under independence) is

$$p_2 = \sum_{i=m(X,A)}^{m(X)} \binom{m(X)}{i} P(A)^i (1 - P(A))^{m(X)-i}. \tag{2}$$

The problem of this measure is that the rules in the same data set cannot be compared. Each rule is tested in a different part of the data, where the antecedent X holds. According to our experiments, the results are also poorer than with Equation 1.

The binomial probability is quite difficult to calculate, but for our purposes it is enough to have an upper bound for the p-value. This guarantees that no rules with a low p-value are lost when the search space is pruned. Additional pruning and ranking can be done afterwards, when the actual binomial probabilities are calculated.

The simplest upper bound is based on the (binomial) *z-score*:

$$\begin{aligned} z(X, A) &= \frac{m(X, A) - \mu_M}{\sigma_M} = \frac{m(X, A) - nP(X)P(A)}{\sqrt{nP(X)P(A)(1 - P(X)P(A))}} \\ &= \frac{\sqrt{n}(\gamma(X, A) - 1)}{\sqrt{\gamma(X, A) - P(X, A)}}. \end{aligned}$$

The z-score measures how many standard deviations (σ_M) the observed frequency $m(X, A)$ deviates from the expected value $\mu_M = nP(X)P(A)$. The corresponding probability can be easily approximated, because z follows the standard normal distribution, when n is sufficiently large and $P(X)P(A)$ (or $1-P(X)P(A)$) is neither close to 0 nor to 1. As a rule of thumb, the approximation can be used, when $nP(X)P(A) \geq 5$ (e.g. (Lindgren, 1993, 147)). According to Carriere (2001), the approximation works well even for $nP(X)P(A) \geq 2$, if continuity correction (subtracting 0.5 from $m(X, A)$) is used.

When $P(X)P(A)$ is low, the binomial distribution is positively skewed. This means that the z-score overestimates the significance. Therefore, we will not use normal approximation to estimate the p-values, but the z-score is used only as a measure (ranking function).

We note that the z-score is not crucial to our method, but several other measure functions can be used, as well. The requirement is that the measure I is a monotonically increasing or decreasing function of $m(X, A)$ and $\gamma(X, A)$. For example, when the expected value $P(X)P(A)$ is very low, we can derive a tight upperbound for p from the *Chernoff bound* (Hoeffding, 1963):

$$P(M > \mu_M(1+\delta)) < \frac{e^{\delta \mu_M}}{(1+\delta)^{(1+\delta)\mu_M}}.$$

By inserting $\delta = \gamma - 1$, where $\gamma = \gamma(X, A)$, and using $\gamma\mu = m(X, A)$, we achieve

$$p_{ch} = P(M > m(X, A)) < \left(\frac{e^{\frac{(\gamma-1)}{\gamma}}}{\gamma} \right)^{m(X,A)} .$$

This is monotonically decreasing with both $m(X, A)$ and γ.

4. Main principles

In this section, we introduce the main principles of the search algorithm. The results are given on such a general level that any suitable measure function or search strategy can be applied.

4.1. Problem definition

Let us first define the problem formally:

Definition 5 (Search problem). Let I be a measure which evaluates the significance of an association rule $X \rightarrow A$. Let us suppose that I is increasing with the goodness of rule (like z) and min_I is the smallest allowed I-value. (For decreasing measures like p and its approximations one has to define max_I.) Given binary data r the problem is to search all association rules $X \rightarrow A$ such that

1. $X \rightarrow A$ expresses a positive correlation, i.e. $\gamma(X \rightarrow A) > 1$,
2. $X \rightarrow A$ is non-redundant, and
3. $I(X \rightarrow A) \geq min_I$.

We note that the user has to select only one parameter, min_I. Alternatively, we could define an optimization problem, where the N best rules (with highest I-values) are searched. However, the tasks are not mutually exclusive. Even if threshold min_I is used, the user can ask to output only N best rules. On the other hand, a loose threshold (e.g. $max_p = 0.05$ or $min_Z = 2.0$) can be used in the optimization problem to prune out rules which cannot be significant.

Let us now assume that we have a measure function $i(fr, \gamma)$ such that for any rule $X \rightarrow A$, $I(X \rightarrow A) = i(P(XA), \gamma(X, A))$. In addition, let i be either monotonically increasing or decreasing with both frequency fr and lift γ. The search problem can be divided into two subproblems:

1. Search all non-redundant rules $X \rightarrow A$ for which $I(X \rightarrow A) \geq min_I$ using i.
2. Calculate the exact p-values and output rules with sufficiently low p.

The postprocessing step is trivial and we will concentrate on only the search step. For simplicity, we assume that i is monotonically increasing; a monotonically decreasing measure function is handled similarly.

4.2. Monotonic and anti-monotonic properties

The key idea of the classical Apriori algorithm (Agrawal and Srikant, 1994; Mannila, Toivonen and Verkamo, 1994) is the anti-monotonicity of frequency. For attribute sets, the monotonicity and anti-monotonicity are defined as follows:

Definition 6 (Monotonic and anti-monotonic properties). Property $\pi : \mathcal{P}(R) \rightarrow \{0, 1\}$ is monotonic, if $(\pi(Y) = 1) \Rightarrow (\pi(X) = 1)$ for all $X \supseteq Y$, and anti-monotonic, if $(\pi(X) = 1) \Rightarrow (\pi(Y) = 1)$ for all $Y \subseteq X$.

When π is anti-monotonic $(\pi(Y) = 0) \Rightarrow (\pi(X) = 0)$ for all $X \supseteq Y$.

Alg. 4.1 The general Apriori algorithm.

```
Input: set of attributes $R$, data set $r$, anti-monotonic property $\pi$
Output: $\{X \subseteq \mathcal{P}(R) | \pi(X) = 1\}$
Method:
  // Initialization
  $S_1 = \{A_i \in R | \pi(A_i) = 1\}$
  $l = 1$
  while $(S_l \neq \emptyset)$
    // Step 1: Candidate generation
    Generate $C_{l+1}$ from $S_l$
    // Step 2: Pruning
    $S_{l+1} = \{c \in C_{l+1} | \pi(c) = 1\}$
    $l = l + 1$
  return $\cup_l S_l$
```

When the measure function defines an anti-monotonic property, the interesting sets or rules can be searched with the general Apriori algorithm (Figure 4.1). The problem is that the measure functions for the statistical significance do not define any anti-monotonic property. However, it turns out that *the upper-bound for the measure function I defines an anti-monotonic property for most set-inclusion relations.*

4.3. Property PS

Let us define the property PS, "potentially significant". Potential significance of set X is a necessary condition for constructing any significant rule $X \setminus B \rightarrow B$.

Definition 7. Let measure function I be as before, min_I a user-defined threshold, and $upperbound(f)$ an upperbound for function f. Let $bestrule(X) = \arg\max_{B \in X}\{I(X \setminus B \rightarrow B)\}$ be the best rule which can be constructed from attributes X. Property $PS : \mathcal{P}(R) \rightarrow \{0,1\}$ is defined as $PS(X) = 1$, iff $upperbound(I(bestrule(X))) \geq min_I$.

Now it is enough to define the conditions under which PS behaves anti-monotonically. The following theorem is the core of the whole search algorithm:

Theorem 1. Let us denote the least frequent attribute in set Z by $minattr(Z) = \arg\min\{P(A_i) \mid A_i \in Z\}$. Let PS, X, and Y be as before. If $(PS(X) = 1)$, then $(PS(Y) = 1)$ for all $Y \subseteq X$ such that $minattr(X) = minattr(Y)$.

Proof. First observe that for all $B \in X$ we have $\gamma(X \setminus B, B)) \leq \frac{1}{P(B)} \leq \frac{1}{P(minattr(X))}$ and $upperbound(I(X \setminus B \rightarrow B)) = i(P(X), \frac{1}{P(minattr(X))})$.

By them we have $upperbound(I(bestrule(X))) = i(P(X), \frac{1}{P(minattr(X))}) \leq i(P(Y), \frac{1}{P(minattr(Y))}) = upperbound(I(bestrule(Y)))$ for all $Y \subseteq X$ such that $minattr(X) = minattr(Y)$.

We have $min_I \leq upperbound(I(bestrule(X)))$ by the definition of $PS(X) = 1$. Hence also $min_I \leq upperbound(I(bestrule(Y)))$, as required for the definition of $PS(Y) = 1$. □

Corollary 1. If $PS(Y) = 0$, then $PS(X) = 0$ for all $X \supseteq Y$ such that $minattr(X) = minattr(Y)$.

We have shown that property PS defines an anti-monotonic property among sets having the same minimum attribute. Let us now consider the exceptional case, when the anti-monotonicity does not hold.

Let us call X an *l-set*, if $|X| = l$. Let the $(l-1)$-subsets, $Y_i \subsetneq X$, $|Y_i| = l-1$, be called X's *parent sets*. Now X has l parent sets, from which only one has a different minimum attribute. The exceptional parent set is $Y_l = X \setminus \{minattr(X)\}$. If $P(minattr(Y_l)) > P(minattr(X))$, Y_l has a lower upperbound for γ than $Y_1, \ldots, Y_{l-1}$ and X have. Therefore, it is possible that Y_l is non-PS, even if X is PS.

4.4. Search strategy

The previous properties define a search strategy, where each set X is classified as

1 potentially significant ($PS(X) = 1$). This means that $Y \supseteq X$ can be significant.

1.1 significant ($Significant(X) = 1$)

1.1.2 redundant ($Red(X) = 1$). The rule is not output. $Y \supsetneq X$ can still be non-redundant and the search continues.

1.1.2 non-redundant ($Red(X) = 0$). The rule is output. $Y \supsetneq X$ can still be more significant and the search continues.

1.1.2.1 minimal ($Minimal(X) = 1$). Now all $Y \supsetneq X$ will be redundant and can be pruned.

1.1.2.2 non-minimal ($Minimal(X) = 0$). The search continues.

1.2 insignificant ($Significant(X) = 0$). The rule is not output. $Y \supsetneq X$ can still be significant and the search continues.

2 not potentially significant ($PS(X) = 0$).

2.1 absolutely non-significant. This means that all $Y \supseteq X$ are insignificant and X cannot occur even as a special parent of PS sets. All $Y \supseteq X$ are pruned.

2.2 special parent. Now X is saved, until the next level is processed.

5. Algorithm

Next, we will describe the StatApriori algorithm. We will first introduce the main idea and then go to details and give the complexity analysis.

5.1. The main idea

The main idea of StatApriori algorithm is given in Algorithm 5.1. The algorithm proceeds in the same manner as the general Apriori (Figure 4.1) except it searches directly rules, not sets. It alternates between the candidate generation and pruning steps, as long as new non-redundant, potentially significant rules can be found. However, special techniques are needed, because property PS is not anti-monotonic in all respects.

First, the attributes are arranged into an ascending order by their frequencies. Let the renamed attributes be $\{A_1, \ldots, A_k\}$, where $P(A_1) \leq \ldots \leq P(A_k)$. The idea is that the candidates are generated in the canonical order. From l-set $X = \{A_1, \ldots, A_l\}$, we can generate $(l+1)$-sets $X \cup \{A_j\}$, where $j > l$. Now all supersets of X have the same upper-bound for the lift, $\gamma \leq \frac{1}{P(A_1)}$. If X is non-PS, then none of its descendants can be PS. Otherwise, we should check the other parent sets $Z \subset X \cup \{A_j\}$, $|Z| = l$. If at most one of them, $(X \setminus \{A_1\}) \cup \{A_j\}$, is non-$PS$, then $X \cup \{A_j\}$ is added to the candidate collection C_{l+1}.

After candidate generation, the exact frequencies are counted from the data. Candidates which are non-PS or can produce only redundant descendants, will be pruned, and others are added to collection S_{l+1}. The minimality of PS rules is checked, because no new candidates are generated from minimal rules. The principles for redundancy and minimality checking are

1. If we have $\gamma(bestrule(X)) = \frac{1}{P(minattr(X))}$, then the lift is already maximal possible, and none of X's specializations can gain a better i-value. The rule is marked as minimal.

Alg. 5.1 The StatApriori algorithm, when the measure function i is monotonically increasing (for a monotonically decreasing function max_I is used instead).

```
Input: set of attributes $R$, data set $r$, threshold $min_I$
Output: $\{X \rightarrow A \mid I(X \rightarrow A) \geq min_I \wedge \neg Red(X \rightarrow A)\}$
Method:
  // Initialization
  order attributes such that $P(A_1) \leq ... \leq P(A_k)$
  $S_1 = \{A_i \in R\}$
  $l = 1$
  while $PS$ non-minimal $l$-sets left
    // Step 1: Candidate generation
    generate $C_{l+1}$ from $S_l$ such that $\forall X \in C_{l+1}$
      1) all $X$'s parent sets are non-minimal, and
      2) first $l$ parents are $PS$
    // Step 2: Pruning
    count frequencies $P(X)$
    $S_{l+1} = \{c \in C_{l+1} | PS(c) = 1\}$
    for $\forall X \in S_{l+1}$
      check the best rule for
        redundancy and minimality
    $l = l + 1$
  output significant, non-redundant rules from $\cup_l S_l$
```

2. If we have $upperbound(I(bestrule(X \cup \{A_j\}))) \leq I(bestrule(Z))$ for some attributes $Z \subsetneq X \cup \{A_j\}$, then $X \cup \{A_j\}$ and all its specializations will be redundant with respect to Z. $X \cup \{A_j\}$ is removed.

5.2. Enumeration tree

The secret of StatApriori is a special kind of enumeration tree, which enables an efficient implementation of pruning principles. A complete enumeration tree lists all sets in the powerset $\mathcal{P}(R)$. In practice, it can be implemented as a trie, where each root–node path corresponds to an item set. StatApriori uses an ordered enumeration tree, where the attributes are arranged into an ascending order by their frequencies. Figure 1 shows an example of a complete enumeration tree, when $R = \{A, B, C, D, E, F\}$, and $P(F) \leq P(E) \leq \ldots \leq P(A)$.

In practice, we have to generate only those parts of the tree which can produce non-redundant significant rules. When no minimum frequency thresholds are used, the final tree contains the globally optimal non-redundant association rules on all root-leaf paths of a complete enumeration tree. If some minimum frequency is required, then we can no more guarantee that the discovered rules are globally optimal. Fortunately, StatApriori is so efficient that there is seldom need for any minimum frequency thresholds.

The following example demonstrates how the enumeration tree is used.

Example 2. Let us simulate the algorithm on the example data given in Table 2. The minimum threshold for the z-score is set to $min_z = 4.00$.

First, the frequencies of single attributes are counted: $P(A) = 0.60$, $P(B) =$

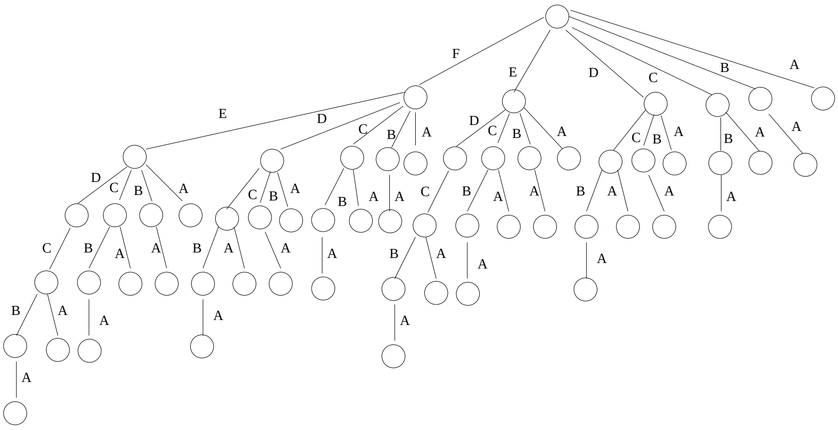

Fig. 1. A complete enumeration tree.

Table 2. Example data.

$m(ABCDEF) = 1$
$m(ABCDE) = 9$
$m(BCDE) = 5$
$m(ABC) = 15$
$m(AC) = 15$
$m(A) = 20$
$m(BDE) = 5$
$m(B) = 20$
$m(F) = 10$
$\Sigma = 100$

0.55, $P(C) = 0.45$, $P(D) = 0.20$, $P(E) = 0.20$, $P(F) = 0.11$. The attributes are inserted to the tree in the ascending order by their frequencies: F, E, D, C, B, A (E and D could be reversed as well).

Figure 2 shows the situation when the second level is processed. All 2-sets in the F-branch have so low frequency that they cannot be PS even with the maximal possible lift $\gamma = P(F)^{-1} = 9.1$. In addition, they cannot occur as special parents for any other sets and they are removed.

All 2-sets in branches E, D, and C are PS. The only significant rule, $E \rightarrow D$, is also minimal, because it has maximal possible lift *in that branch*, $\gamma = P(D)^{-1} = P(E)^{-1} = 5.00$. The last 2-set BA in non-PS, because it cannot produce a significant rule even with its maximal possible lift $\gamma = P(B)^{-1} = 1.8$. However, it can be a special parent of another rule, and it is saved.

Figure 3 shows the situation at the third level. Since ED was already minimal, it is not expanded. All generated 3-sets are PS, but only one rule, $BA \rightarrow C$, is significant. This is also an example, where a special parent was needed. Since the rule is minimal, no 4-sets are generated and the search finishes.

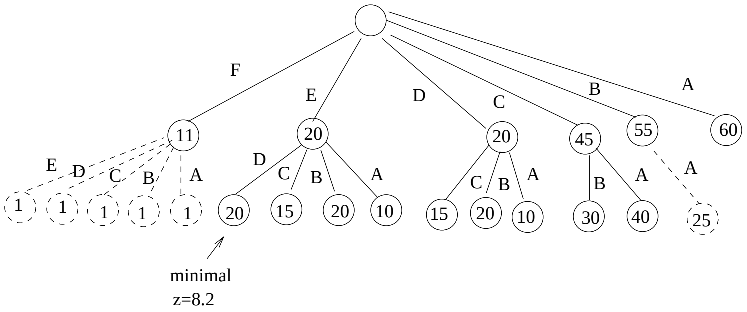

Fig. 2. Example when level 2 is processed.

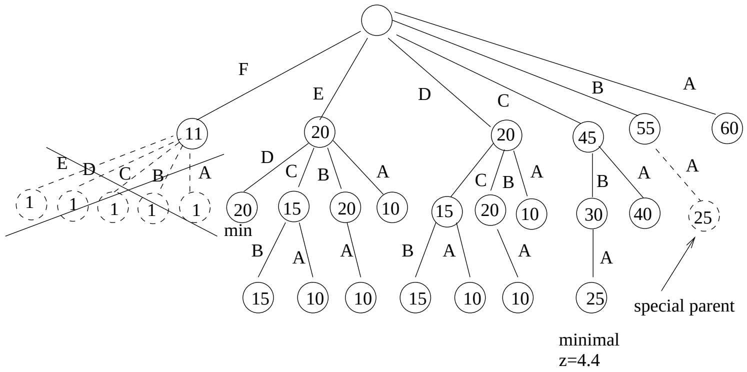

Fig. 3. Example when level 3 is processed.

5.3. Handling special parents

The previous example demonstrated that some non-PS l-sets are needed when $l+1$-sets are checked. This situation occurs when $bestrule(X) = Y \rightarrow B$, $X = YB$, is PS, but Y is non-PS. In practice, there are two way to handle the situation:

1. Only PS sets are stored into the tree. A new $(l+1)$-set is generated, if its l first parent sets are in the tree (i.e. PS). If the last parent was missing, it is added temporarily for rule checking and also its frequency is counted.
2. Non-PS sets are divided into two categories. If an l-set cannot be even a special parent, it is deleted, but otherwise it is stored into the tree. After the lth level has been processed, all useless non-PS sets are deleted.

The current implementation uses the latter approach, which has proved to be more efficient. Potential special parents can be identified by two principles:

Principle 1. Let $A_{min} = minattr(R)$ be the minimum attribute in the whole data set. The maximal possible lift is $P(A_{min})^{-1}$. If a non-PS set X is non-PS with this lift then it cannot be even a special parent. I.e. if $i(P(X), P(A_{min})^{-1}) < min_I)$, then X can be deleted.

Principle 2. Let Y be a non-PS l-set in branch A_i (i.e. $mina(Y) = A_i$). Y

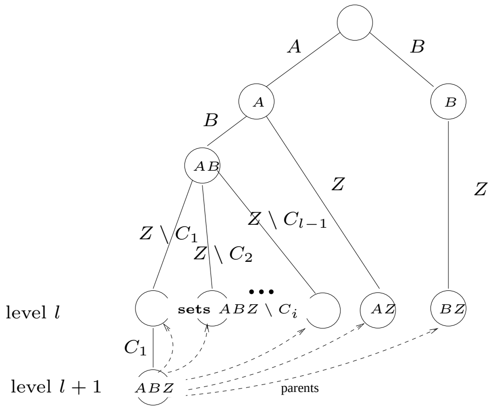

Fig. 4. When BZ is non-PS, set ABZ cannot be PS, if any of $ABZ \setminus \{C_i\}$ or AZ is non-PS.

cannot be a special parent for any $(l+1)$-set, unless some of the previous branches $A_1, ..., A_{i-1}$ contain at least l PS, non-minimal l-sets. The reason is that each $(l+1)$-set has its first l parents in the same main branch and only the last parent set is in one of the succeeding branches.

The second principle is so important that we give a detailed proof:

Theorem 2. Let set ZB, where $B = minattr(ZB)$, $|Z| = l - 1$, be non-PS. Then BZ cannot be a special parent of any set, if for all A_i, $P(A_i) \leq P(B)$, the number of PS $(l-1)$-sets in branch A_i, $S_{l-1}^{A_i} = |\{A_iY \mid PS(A_iY)\}| < l$.

Proof. Let $A = A_i$, $P(A) \leq P(B)$. Let us first consider $(l+1)$-sets, whose special parents BZ could be. In branch A, the only possible set is ABZ. Its other l parents are $ABZ \setminus \{C_i\}$, $i = 1, ..., l-1$, and AZ. If $S_{l-1}^{A} < l$, then either some $ABZ \setminus \{C_i\}$ or AZ must be non-PS, and ABZ cannot be PS based on Corollary 1. The idea is shown in Figure 4. Notice that the Figure represents only the nodes needed to decide if ABZ is PS. Therefore, all nodes on the left (ABZ's left sisters, $ABZ \setminus \{D_1\}$'s left sisters, and A's left sisters) are missing.

Let us now consider $(l+m+1)$-sets, whose special parent some BZ's descendant could be. Since BZ was non-PS, all its descendants (sets BZQ, where $minattr(BZQ) = B$) are also non-PS.

For any Q, $|Q| = m \geq 1$, the only possible set in branch A having BZQ as its special parent is $ABZQ$. Its other $l+m$ parents are $ABZ \setminus \{C_i\}Q$, $i = 1, ..., l-1$, $ABZQ \setminus \{D_i\}$, $i = 1, ..., m$, and AZQ. If any of sets $ABZ \setminus \{C_i\}$ was non-PS, then also $ABZ \setminus \{C_i\}Q$ is non-PS. On the other hand, if AZ was non-PS, then AZQ is non-PS. In either case $ABZQ$ cannot be PS. The idea is shown in Figure 5. □

At each level, the first principle is used, until some branch contains too few PS, non-minimal sets and we can change to the second principle. In practice, the second principle is much more effective, since the first branches contain the least frequent minimum attributes and are the reason for over-optimistic upperbounds for the lift.

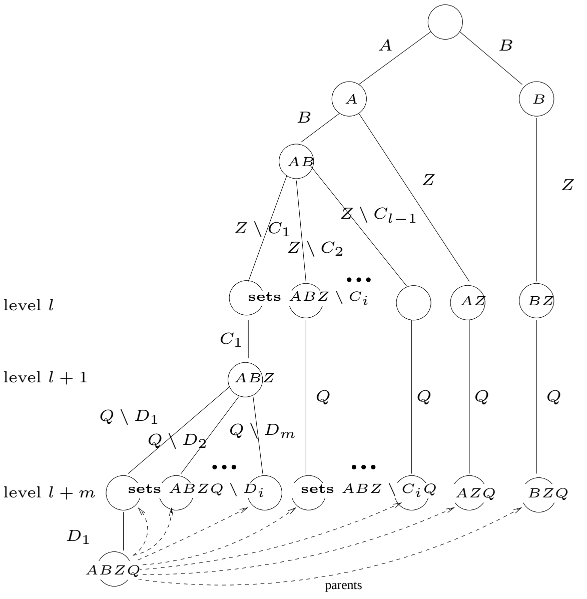

Fig. 5. When BZ is non-PS, set $ABZQ$ cannot be PS, if any of $ABZ \setminus \{C_i\}$ or AZ is non-PS.

5.4. Frequency counting

Frequency counting is the main bottle-neck of association rule discovery algorithms. In the traditional Apriori (and also the first implementation of StatApriori), the set of all l-candidates are checked at once. It is very space consuming, when the number of candidates can easily be millions. When each candidate is checked immediately, useless ones can also be discarded immediately. In practice, the saving of space can be crucial for the execution without any minimum frequency thresholds.

Another aspect is whether to use horizontal or vertical data layout. In the horizontal layout each row contains a list of attributes which occur on the row. In practice, this is usually implemented as n bitvectors of size k. In the vertical layout the roles of rows and attributes are reversed. Each attribute is associated with a list of rows, where the attribute is true. Now the data set can be implemented as k bitvectors of size n. In practice, the vertical layout is more efficient and it is used also in the current implementation of StatApriori.

Let us now analyze the time requirement for frequency counting, when data is stored into bitvectors. Each bitvector is implemented by $\frac{m}{c}$ integers, where m is the length of the bit vector and constant c is the number of bits in the machine word (e.g. 32 or 64, depending on the architecture). Each integer can be checked fast by bitwise logical operations. In practice, we can consider the bitwise logical operations on integers as atomic operations which take constant time.

When the horizontal layout is used, counting the frequency of each l-set requires in the worst case $n \cdot \min\{l, \frac{k}{c}\}$ time steps. If $l < k/c$, the attributes of an l-set can occur in at most l integers, and checking one row takes at most l steps. The checking is repeated on all n rows.

In the vertical layout, the frequency of set $X = A_1 ... A_l$ is the same as the number of 1-bits in the bitwise AND of vectors $v(A_1), ..., v(A_l)$. This observation was made already 1996 by Yen and Chen (1996), but in that time the memory sizes were so small that the technique could not be utilized for large data sets. Counting the frequency of one l-set takes $\frac{n}{c}(l+1)$ time steps, which consist of $\frac{nl}{c}$ bitwise AND operations and counting the number of 1-bits in the result ($\frac{n}{c}$ steps) on each of l rows.

When we compare the requirements we see that the vertical layout is in practice more efficient, even if the time complexities are the same: $\frac{n}{c}(l+1) < n \cdot \min\{\frac{k}{c}, l\}$ for all $l < k-1$.

5.5. Time complexity

It is known that the problem of searching all frequent attribute sets is NP-hard, in the terms of the number of attributes, k (Jermaine, 2005). The worst case happens, when the most significant association rule involves all k attributes, and all 2^k attribute sets are generated. The worst case complexity of the algorithm is $O(\max\{k^2, nk\}2^k)$. Usually, when $k < n$, this reduces to $O(n^2 2^k)$.

Theorem 3. The worst-case time complexity of StatApriori is $O(\max\{k^2, nk\}2^k)$, where n is the number of rows and k is the number of attributes.

Proof. The initialization (generation of 1-sets) takes $n \cdot k$ steps. Producing l-sets and their best rules takes $l^2|C_l| + |C_l|n \cdot l + 2|S_l|l$ time steps.

The first term is the time complexity of the candidate generation. Each candidate has l parents and each parent can be found (using a trie) in $l-1$ steps.

The second term is the complexity of the frequency counting. For each of $|C_l|$ candidates, we have to perform l bitwise AND operations on bitvectors of length n and count the number of 1-bits in the results. For one candidate set the complexity is $O(n \cdot l)$ (see also subsection 5.4).

The third term is the complexity of the rule selection phase: for each of $|S_l|$ sets, all l parents are checked. Checking is done at most twice: once for calculating the maximal I-value (selecting the best rule) and second time for checking the redundancy. Each checking can be implemented in constant time, if the parent pointers are stored into a temporary structure in candidate generation phase.

Since $|C_l| \geq |S_l|$, the total complexity is

$$\sum_{l=2}^{k} \max\{l^2, nl\}|C_l| \leq \max\{k^2, nl\} \sum_{l=2}^{k} \binom{k}{l} = O(\max\{k^2, nk\}2^k).$$

□

In practice, we never have to scan the whole search space, because all k-sets cannot express non-redundant significant positive dependencies. However, it is difficult to prove tighter upperbounds or estimate the maximal number of non-redundant significant rules.

Table 3. Description of data sets and test parameters (n is the size of the learning set)

Data			**StatApriori**		**Apriori**
	n	k	min_Z	zfr	min_{fr}
Mushroom	5416	120	40	0.0015	0.22
Chess	2130	75	18	0.00058	0.75
T10I4D100K	66581	1000	150	0.000035	0.001
T40I10D100K	66581	1000	70	0.014	0.013
Plants	15088	70	70	0.015	0.12
Accidents	226788	470	260	0.000016	0.45
Pumsb	32697	8000	170	0.00089	0.80
Retail	58774	18000	150	0.00011	0.70

6. Experiments

The main goal of the experiments was to evaluate the speed–accuracy ratio of the StatApriori algorithm. Even a clever algorithm is worthless unless it can produce better results or perform faster than the existing methods. It was expected that StatApriori cannot compete in speed with the traditional methods, but instead it is likely to produce more accurate rules.

6.1. Test setting

In the experiments, StatApriori was compared to the traditional Apriori, followed by postprocessing where non-redundant, optimal rules were selected. For the traditional Apriori, we used an efficient prefix-tree implementation by C. Borgelt (Borgelt and Kruse, 2002). The tested measures were χ^2, J-measure (Smyth and Goodman, 1992), z-score, and frequency. The z-score was included here, too, to test the effect of frequency-based pruning on results. In the StatApriori we used mainly the z-score, but the tails of the distribution (when $nP(X)P(A) \leq 1$) were also checked using upper and lower bounds for the logarithm of binomial probability. The best rules were selected by an upperbound for $log(p)$, because calculating exact p would have caused an underflow in most data sets.

The data sets and test parameters are described in Table 3. All data sets except *Plants* are classical benchmark data sets from the FIMI repository (*Frequent Itemset Mining Dataset Repository*). Some of the data sets are originally from the Machine Learning repository (Asuncion and Newman, 2007), but in FIMI they have already been binarized and transformed into transactional form. *Plants* lists all plant species growing in the U.S.A. and Canada. Each row contains a species or genus and lists the states where it grows. The data has been extracted from the USDA plants database (*The PLANTS Database*, 2008). All data sets and the source code can be accessed on `http://www.cs.helsinki.fi/u/whamalai/datasets.html`.

Table 3 lists also the minimum z-scores used by StatApriori and the minimum frequency threshold used by traditional Apriori. In StatApriori we did not use any minimum frequency thresholds, except requirement $m(X, A) \geq 5$ (implemented in the program), which is the minimum requirement for performing any statistical tests. In addition, we report value zfr, the minimum frequency threshold corresponding to the z-score. The same results could be achieved by the traditional Apriori, if zfr could be used.

The minimum frequency thresholds for traditional Apriori were set as low as possible. The large thresholds are partly due to heavy postprocessing phase. For feasibility, the thresholds were set to avoid over 500 000 rules. However, the dense data sets are difficult for Apriori even without this restriction. For example, Apriori cannot handle *Chess* with $min_{fr} < 0.50$. *Retail* was so heavy that we had to restrict the length of the antecedent to 4 attributes. In StatApriori no restrictions were needed.

All experiments were executed on Intel Core Duo processor T5500 1.66GHz with 1 GB RAM, 2MG cache, and Linux operating system.

6.2. Evaluation measures

Since association rules can be used for both predictive and descriptive modelling, both aspects were assessed. When association rules are used for prediction, the natural measure is the average prediction error in the test set:

$$err = \frac{1}{n_T} \sum_{m(X)>0} P(\neg A|X),$$

where $n_T = |\{X \rightarrow A \mid m(X) > 0 \text{ in the test set}\}|$ is the number of rules which could be applied in the test set. Now it is important that the test sets are sufficiently large, because otherwise the mean error is based on just a few (most common) rules. Therefore, we saved 1/3 of each data set for testing, and used 2/3 for “learning” (searching the rules). Since variation among different divisions to test and learning sets can be relatively large, all experiments were repeated ten times on random divisions to learning and test sets.

The most important factor for the prediction quality is high confidence. However, the estimate of the confidence in the learning set should be accurate, i.e. the confidence in the test set should be at least equally high. In addition, the lift should be $\gamma > 1.0$ to improve the accuracy compared to expectation $err = P(\neg A)$. Otherwise the rule is useless and in the worst case only decreases the accuracy ($\gamma < 1.0$).

When association rules are used for descriptive purposes, the prediction accuracy is not so important, but instead we would like to find strong dependencies. This can be measured by the average lift

$$avg(\gamma) = \frac{1}{n_T} \sum_{m(X)} \gamma(X \rightarrow A).$$

We note that $\gamma(X \rightarrow A) = 1$, if X occurs in the test set but A does not.

Once again the estimate of lift in the learning set should also hold in the test set. Therefore, we calculated the z-score for the difference of the lift distribution (characterized by the mean and variance) between the learning set and the test set:

$$z(\Delta\gamma) = \frac{\mu(\gamma_L) - \mu(\gamma_T)}{\sqrt{\frac{\sigma^2(\gamma_L)}{n_L} + \frac{\sigma^2(\gamma_T)}{n_T}}},$$

where γ_L is the lift variable in the learning set and γ_T in the test set, μ and σ^2 notate the mean and variance, n_L is the number of rules considered in the

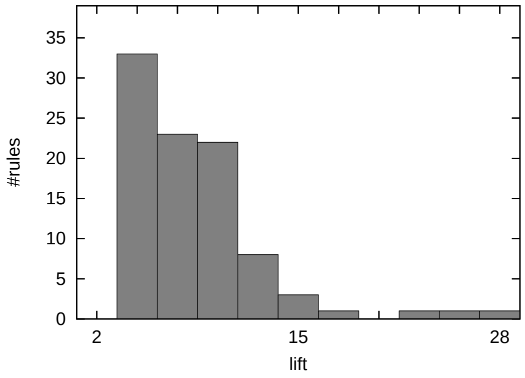

Fig. 6. The lift distribution among the 100 best rules by fr measure in a learning set of T40I10D100K.

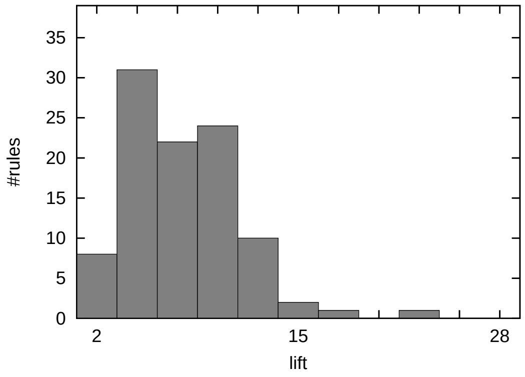

Fig. 7. The lift distribution among the 100 best rules by fr measure in a test set of T40I10D100K.

learning set, and n_T is the number of rules, which could be applied in the test set. The closer the z-score is to zero, the more similar the distributions are. If the score is negative, then the correlations are stronger in the test set than in the learning set.

For example, Figures 6 and 7 show the lift distribution in a learning and test set of data set T40I10D100K, when the rules were selected by frequency. Now $z(\Delta\gamma) = -0.0$, indicating no significant difference. The score is relatively conservative, and the difference should be quite large, before the score deviates from 0.

As a final test, we calculated the proportions of useful and harmful rules. When the goal is to find positive dependencies, all rules expressing negative dependencies in the future data are considered harmful. On the other hand, all rules expressing positive dependencies in the future data are useful. Rules which express independence in the future data are quite harmless, but they do not offer any new information.

Both useful and harmful rules were divided into two categories. Rule $X \rightarrow A$ was classified as *slightly useful*, if

$P(X)P(A) < P(XA) < P(X)P(A) + \sqrt{nP(X)P(A)(1 - P(A)P(X))}$

in the test data. I.e. the rule expresses positive dependency but the frequency deviates at most one standard deviation from the expectation. If the deviation was from expectation was more than one standard deviation, the rule was considered *useful*.

Similarly,

$X \rightarrow A$ was classified as *slightly harmful*, if

$P(X)P(A) > P(XA) > P(X)P(A) - \sqrt{nP(X)P(A)(1 - P(A)P(X))}$

in the test data. Now the rule expresses negative dependency but the frequency deviates at most one standard deviation from the expectation. If the deviation was larger, the rule was considered *harmful*.

6.3. Results

The results are given in Tables 4, 5, 6, 7, and 8. The average values among all data sets are given on the last row. Proportions of useful and harmful rules are shown in Figures 8 and 8.

Let us first analyze the average prediction error, when the minimum confidence is $min_{cf} = 0.90$. If look at only the average prediction error among all data sets, StatApriori looks to have performed slightly worse than other approaches. The average prediction error was 0.08, while the best measures with the traditional Apriori, J and fr produced average error 0.05. However, this is only part of the truth. Both J and especially fr favoured trivial rules with lift value near 1.0. Frequency measure produced lift $\gamma \leq 1.1$ in five data sets from eight, and J measure in three data sets from eight. It means that equally good predictions could have been done without the rules.

If StatApriori was run with larger min_Z values, it also produces smaller prediction errors (in fact smaller than other approaches). However, the number of discovered rules was also smaller. Therefore, we used smaller min_Z values to find at least 100 significant rules like with other measures.

The most difficult data set for StatApriori was *Chess*, where we also had to use very low min_Z to find enough rules. When we analyzed the data set we found that the most significant dependencies were in fact negative, and therefore not detected by StatApriori. On the other hand, the minimum frequency threshold used by traditional Apriori was very high ($min_{fr} = 0.75$), meaning that both the consequent and antecedent are very common. Then it is only likely that they occur together, even if they were dependent.

It is interesting that χ^2, J and z produced smaller prediction error, when the minimum confidence was $min_{cf} = 0.60$ than with $min_{cf} = 0.90$. The rules with $min_{cf} = 0.90$ were also more complex (longer) and only about 80% of them could be applied to the test set.

When we look at the average lift values among all data sets, the situation is quite different. Measures which produced smallest prediction errors, produced also smallest lift values. *StatApriori* discovered the strongest dependencies with both $min_{cf} = 0.6$ and $min_{cf} = 0.9$. This is not surprising since it was designed to search statistically significant positive dependencies. The second best method was χ^2 with the traditional Apriori (also designed for detecting statistical dependencies), and the last was pure frequency measure.

One reason for small lift values in the traditional Apriori is the need for

Table 4. Results of StatApriori when $min_{cf} = 0.60$ and $min_{cf} = 0.90$. (Average lift, rule accuracy, and the difference of lift.)

		0.60			0.90	
	γ	*err*	$z(\Delta)$	γ	*err*	$z(\Delta)$
Mushroom	4.8	0.03	-1.54	4.83	0.03	-1.54
Chess	45.4	0.33	1.4	96.5	0.15	-0.81
T10I4D100K	102.8	0.11	-0.16	124.4	0.07	-0.05
T40I10D100K	36.9	0.06	-0.35	38.2	0.05	-0.35
Plants	5.47	0.15	0.22	5.5	0.15	0.15
Accidents	1506.5	0.23	0.02	1839.0	0.10	0.57
Pumsb	61.3	0.10	-0.16	90.1	0.03	-0.84
Retail	133.0	0.02	-0.30	133.0	0.02	-0.30
avg	237.0	0.13	-0.11	291.4	0.08	-0.40

Table 5. Results of traditional Apriori with χ^2 measure when $min_{cf} = 0.60$ and $min_{cf} = 0.90$. (Average lift, rule accuracy, and the difference of lift.)

		0.60			0.90	
	γ	*err*	$z(\Delta)$	γ	*err*	$z(\Delta)$
Mushroom	2.8	0.15	-2.75	2.9	0.05	-1.75
Chess	1.0	0.09	0.16	53.4	0.12	-1.31
T10I4D100K	754.8	0.09	-1.67	81.6	0.11	-0.16
T40I10D100K	48.8	0.06	-11.81	47.9	0.03	-4.76
Plants	5.2	0.21	-0.13	5.3	0.08	0.07
Accidents	1.1	0.22	0.03	1.1	0.04	-0.02
Pumsb	1.1	0.06	-0.53	1.1	0.06	-0.52
Retail	1.1	0.07	0.10	1.1	0.06	0.09
avg	102.0	0.12	-2.08	86.2	0.07	-1.22

Table 6. Results of traditional Apriori with J measure when $min_{cf} = 0.60$ and $min_{cf} = 0.90$. (Average lift, rule accuracy, and the difference of lift.)

		0.60			0.90	
	γ	*err*	$z(\Delta)$	γ	*err*	$z(\Delta)$
Mushroom	2.7	0.02	-0.74	2.8	0.03	-1.06
Chess	1.0	0.03	0.01	1.0	0.03	-0.00
T10I4D100K	81.6	0.11	-0.16	79.4	0.20	-1.51
T40I10D100K	30.0	0.04	0.27	30.0	0.04	0.36
Plants	4.3	0.07	0.00	4.4	0.05	0.06
Accidents	1.2	0.13	0.01	1.1	0.03	-0.02
Pumsb	1.1	0.03	-0.19	1.1	0.03	-0.22
Retail	1.1	0.02	-0.10	1.2	0.01	-0.15
avg	15.4	0.06	-0.11	15.1	0.05	-0.32

Table 7. Results of traditional Apriori with z measure when $min_{cf} = 0.60$ and $min_{cf} = 0.90$. (Average lift, rule accuracy, and the difference of lift.)

		0.60			0.90	
	γ	*err*	$z(\Delta)$	γ	*err*	$z(\Delta)$
Mushroom	2.8	0.07	-2.55	2.9	0.04	-1.63
Chess	1.0	0.08	0.37	1.0	0.06	0.41
T10I4D100K	455.5	0.05	0.81	364.3	0.22	-2.44
T40I10D100K	48.7	0.04	15.81	47.7	0.03	-4.81
Plants	5.3	0.14	0.11	5.2	0.07	0.27
Accidents	1.1	0.20	-0.01	1.1	0.04	-0.01
Pumsb	1.1	0.05	-0.55		0.05	-0.75
Retail	1.1	0.06	0.20	1.1	0.06	0.14
avg	64.6	0.09	1.77	53.0	0.07	-1.10

Table 8. Results of traditional Apriori with fr measure when $min_{cf} = 0.60$ and $min_{cf} = 0.90$. (Average lift, rule accuracy, and the difference of lift.)

		0.60			0.90	
	γ	*err*	$z(\Delta)$	γ	*err*	$z(\Delta)$
Mushroom	1.1	0.09	0.02	1.1	0.03	0.07
Chess	1.0	0.03	0.16	1.0	0.03	0.16
T10I4D100K	37.4	0.14	-0.62	37.4	0.14	-0.62
T40I10D100K	10.1	0.07	-0.05	10.1	0.07	-0.05
Plants	4.0	0.06	-0.10	4.0	0.06	-0.10
Accidents	1.0	0.11	0.00	1.0	0.02	0.00
Pumsb	1.0	0.02	0.00	1.0	0.02	0.00
Retail	1.0	0.02	0.00	1.0	0.02	0.00
avg	6.7	0.12	0.02	7.1	0.05	-0.07

large minimum frequency thresholds. For any rule the upperbound of the lift is $\gamma \leq \frac{1}{min_{fr}}$.

When we look at $z(\Delta(\gamma))$-values, we see that *StatApriori* produced the most stable results, in the sense that the lift distributions were nearly identical in the learning and test sets. J-measure and frequency produced also quite stable results, but in the case of frequency, it only means that rules were spurious (or trivial) both in the learning and the test set.

Proportions of useful and harmful rules give a good summary of the quality of rules. Since the requirement for useful rules was relatively loose, most measures performed well. The only exception was pure frequency, which does not measure statistical dependence at all. However, in sparse data sets (typical market basket data) the most frequent rules often happen to express statistical dependencies, too. Set *Chess* was the most problematic for traditional Apriori with all measures. The likely reason is that *Chess* contains relatively many negative dependencies. We note that the redundancy reduction did already improve the results and without it the ratio of useful and harmful rules was even worse.

The average execution times of StatApriori are listed in Table 9. In most cases

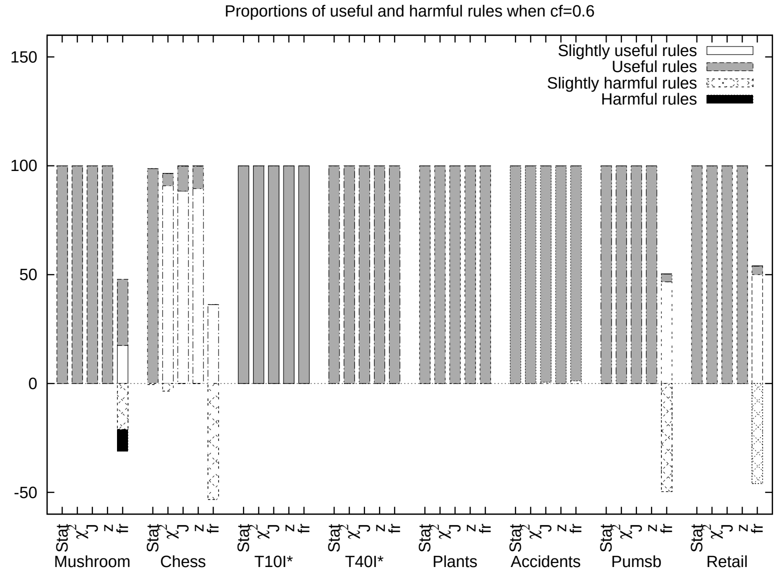

Fig. 8. Proportions of useful and harmful rules among 100 best rules, when $min_{cf} = 0.6$.

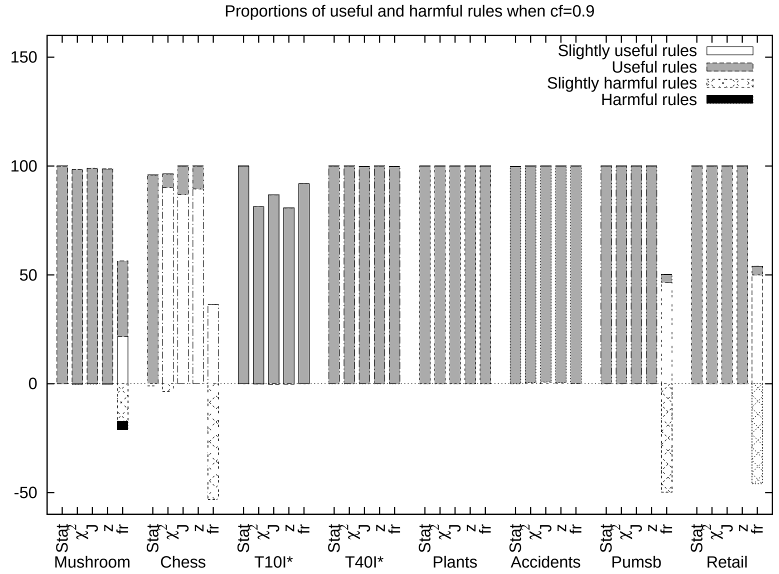

Fig. 9. Proportions of useful and harmful rules among 100 best rules, when $min_{cf} = 0.9$.

Table 9. Average execution times (sec) by StatApriori.

Mushroom	1
Chess	37
T10I4D100K	8
T40I10D100K	12
Plants	6
Accidents	6
Pumsb	31
Retail	62

StatApriori performed faster than the traditional Apriori, even if no minimum frequency thresholds were used.

7. Related research

In the previous research, statistical measures have been used mostly in the post-processing phase. Relatively few algorithms have used them to guide the search. Usually, the task has been to search all significant classification rules $X \to C$, with a fixed attribute C. In this case, it is also desirable that the attributes, rather than events, are dependent, and also negations are considered. As a result, the methods are not scalable for exhaustive search of all significant association rules.

The most commonly used measure is χ^2, which was used already by Brin et al. (Brin, Motwani and Silverstein, 1997; Silverstein et al., 1998) to find the "minimally correlated sets", i.e. the most general attribute sets X such that $\chi^2(X)$ is sufficiently high. No rules were derived, and in fact it is possible that none of the association rules generated from a correlated set is necessarily significant (Morishita and Nakaya, 2000).

Morishita et al. (Morishita and Nakaya, 2000; Morishita and Sese, 2000) developed a branch-and-bound method, which utilizes the convexity of the χ^2 function. The search was restricted to classification rules with a fixed consequent C.

Nijssen and Kok (Nijssen and Kok, 2006) searched also classification rules with χ^2 and some other functions. They defined a minimum frequency threshold from threshold min_{χ^2} and searched the rules with the traditional Apriori. They found that the approach was quite inefficient, because the resulting minimum frequency thresholds were too low.

Other researchers have developed methods for searching association rules with zero or extremely low minimum frequency thresholds. If this could be done efficiently, the significant association rules could be selected afterwards, in the post-processing phase.

Cohen et al. (Cohen, Datar, Fujiwara, Gionis, Indyk, Motwani, Ullman and Yang, 2001) and Fujiwara et al. (Fujiwara, Ullman and Motwani, 2000) searched only simple rules of form $A \to B$. Wang et. al (Wang, Zhou and He, 2000; Wang, He and Cheung, 2001) searched classification rules with a minimum confidence threshold. The algorithm was tested only with one small data set and the scalability remained unknown.

Poor scalability has been the main problem of apriori-type of algorithms, too.

Often the running times are not reported, but the algorithm has been tested only with small data sets which hints that it does not scale up well.

Koh and Pears (Koh, 2008; Koh and Pears, 2007) implemented an interesting variation of the standard Apriori, where Fisher's exact test was used as a significance criterion. Unfortunately, only a few experiments were reported. Fisher's exact test was used also in (Koh, Rountree and O'Keefe, 2008) to find "sporadic rules", i.e. strong and statistically significant rules on rare consequent attributes.

ORD algorithm by Li (2006) is maybe the closest to our algorithm. ORD is a general Apriori-type algorithm, which searches optimal, non-redundant classification rules with some interestingness measure i (e.g. confidence or lift). No minimum frequency threshold is needed, but instead min_I and *minimum local support* σ (threshold for $P(X|A)$) are used.

A difference to our approach is that Li defined redundancy in a weaker sense (i.e. less pruning) than we do, by comparing rule $X \rightarrow A$ to all more general rules with the same consequent, i.e. $Y \rightarrow A$, $Y \subsetneq X$. It also means that Li's efficient pruning principle ("closure property") stating that if $P(XY) = P(X)$, then all rules $XYQ \rightarrow C$ are redundant, cannot be applied in StatApriori. The reason is that even if $\gamma(XYQ \rightarrow C) \leq \gamma(XQ \rightarrow C)$, there can still be more significant rules (with a higher lift) of form $\gamma(XYQ \setminus B \rightarrow B) > \gamma(XQ \setminus B \rightarrow B)$.

The scalability of ORD algorithm remained open, since it was tested only with relatively small data sets and fixed rule length.

In addition to search algorithms, there has been a lot of research and discussion on related topics like problems of frequency-confidence framework (Berzal et al., 2001; Aggarwal and Yu, 1998; Webb, 2006; Webb, 2007), how to estimate the confidence in a statistically sound way (Mozina, Demsar, Zabkar and Bratko, 2006), or solve the multiple testing problem (Webb, 2007).

8. Conclusions

Searching statistically significant association rules is an important but neglected problem. So far, it has been considered computationally infeasible with any larger data sets. In this paper, we have shown that its is possible to search all statistically significant rules in a reasonable time. We have introduced a set of effective pruning properties and a breadth-first search strategy, StatApriori, which implements them.

StatApriori can be used in two ways: either to search K most significant association rules or all rules passing the given significance threshold (minimum z-score). This enables the user to solve the multiple testing problem (i.e. setting the significance threshold) in a desired way or use the algorithm only for ranking the most significant rules.

In the same time, StatApriori solves another important problem, and prunes out all redundant association rules. According to experimental results, this improves the rule quality by avoiding overfitting. Together, the z-score and redundancy reduction provide a sound method for rule discovery. I.e. the discovered rules have a high probability to hold in the future data.

The current implementation of StatApriori is very efficient and in the experiments all data sets could be searched without any minimum frequency thresholds. However, if we want to find more than 100 best rules in dense data sets, the search can become slow. This could be solved by implementing the pruning principles in a depth-first search.

In the future research, we are going to implement other measure functions like J-measure and χ^2 (for positive dependencies). It would also be interesting to compare the results with randomization test approach. In addition, we are going to apply the basic idea of StatApriori for searching both positive and negative dependencies. The current method cannot be used as such, but we have already invented new pruning principles. The final goal is to develop an efficient algorithm for searching the most significant, general association rules, containing any propositional logic formulas.

Acknowledgements. This work was supported by a grant from the Finnish Concordia Fund.

References

Aggarwal, C. and Yu, P. (1998), A new framework for itemset generation, *in* 'Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1998)', ACM Press, New York, USA, pp. 18–24.

Agrawal, R., Imielinski, T. and Swami, A. (1993), Mining association rules between sets of items in large databases, *in* P. Buneman and S. Jajodia, eds, 'Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data', Washington, D.C., pp. 207–216.

Agrawal, R. and Srikant, R. (1994), Fast algorithms for mining association rules, *in* 'Proceedings of the 20th International Conference on Very Large Data Bases, VLDB'94', Morgan Kaufmann, pp. 487–499.

Agresti, A. and Min, Y. (2005), 'Frequentist performance of Bayesian confidence intervals for comparing proportions in 2 × 2 contingency tables', *Biometrics* **61**, 515–523.

Asuncion, A. and Newman, D. (2007), 'UCI machine learning repository'.
URL: *http://www.ics.uci.edu/∼mlearn/MLRepository.html*

Bastide, Y., Pasquier, N., Taouil, R., Stumme, G. and Lakhal, L. (2000), Mining minimal non-redundant association rules using frequent closed itemsets, *in* 'Proceedings of the First International Conference on Computational Logic (CL'00)', Vol. 1861 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 972–986.

Benjamini, Y. and Hochberg, Y. (1995), 'Controlling the false discovery rate: A practical and powerful approach to multiple testing', *Journal of the Royal Statistical Society, Series B* **57**(1), 289–300.

Berzal, F., Blanco, I., Sánchez, D. and Miranda, M. A. V. (2001), A new framework to assess association rules, *in* 'Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis (IDA'01)', Vol. 2189 of *Lecture Notes In Computer Science*, Springer-Verlag, London, UK, pp. 95–104.

Borgelt, C. and Kruse, R. (2002), Induction of association rules: Apriori implementation, *in* 'Proceedings of the 15th Conference on Computational Statistics (COMPSTAT 2002)', Physica Verlag, Heidelberg, Germany.

Brin, S., Motwani, R. and Silverstein, C. (1997), Beyond market baskets: Generalizing association rules to correlations, *in* J. Peckham, ed., 'Proceedings ACM SIGMOD International Conference on Management of Data', ACM Press, pp. 265–276.

Carriere, K. (2001), 'How good is a normal approximation for rates and proportions of low incidence events?', *Communications in Statistics: Simulation and Computation* **30**, 327–337.

Cohen, E., Datar, M., Fujiwara, S., Gionis, A., Indyk, P., Motwani, R., Ullman, J. and Yang, C. (2001), 'Finding interesting associations without support pruning', *IEEE Transactions on Knowledge and Data Engineering* **13**(1), 64–78.

Freedman, D., Pisani, R. and Purves, R. (2007), *Statistics*, 4th edn, Norton & Company, London.

Frequent Itemset Mining Dataset Repository (n.d.). Retrieved 10.2. 2009.
URL: *http://fimi.cs.helsinki.fi/data/*

Fujiwara, S., Ullman, J. and Motwani, R. (2000), Dynamic miss-counting algorithms: Finding implication and similarity rules with confidence pruning, *in* 'Proceedings of the 16th International Conference on Data Engineering, (ICDE'00)', IEEE Computer Society, pp. 501–511.

Gionis, A., Mannila, H., Mielikäinen, T. and Tsaparas, P. (2007), 'Assessing data mining results via swap randomization', *ACM Transactions on Knowledge Discovery from Data* **1**(3), 14:1–14:32.

Hämäläinen, W. and Nykänen, M. (2008), Efficient discovery of statistically significant association rules, *in* 'Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008)', pp. 203–212.

Hoeffding, W. (1963), 'Probability inequalities for sums of bounded random variables', *Journal of the American Statistical Association* **58**, 13–30.

Jermaine, C. (2005), 'Finding the most interesting correlations in a database: how hard can it be?', *Information Systems* **30**(1), 21–46.

Koh, Y. and Pears, R. (2007), Efficiently finding negative association rules without support threshold, *in* 'AI 2007: Advances in Artificial Intelligence, Proceedings of the 20th Australian Joint Conference on Artificial Intelligence (AI 2007)', Vol. 4830 of *Lecture Notes in Computer Science*, Springer, pp. 710–714.

Koh, Y., Rountree, N. and O'Keefe, R. (2008), 'Mining interesting imperfectly sporadic rules', *Knowledge and Information Systems* **14**(2), 179–196.

Koh, Y. S. (2008), Mining non-coincidental rules without a user defined support threshold, *in* 'Advances in Knowledge Discovery and Data Mining, Proceedings of the 12th Pacific-Asia Conference (PAKDD 2008)', Vol. 5012 of *Lecture Notes in Computer Science*, Springer, pp. 910–915.

Li, J. (2006), 'On optimal rule discovery', *IEEE Transactions on Knowledge and Data Engineering* **18**(4), 460–471.

Lindgren, B. (1993), *Statistical Theory*, 4th edn, Chapman & Hall, Boca Raton, U.S.A.

Liu, G., Li, J. and Wong, L. (2008), 'A new concise representation of frequent itemsets using generators and a positive border', *Knowledge and Information Systems* **17**(1), 35–56.

Mannila, H., Toivonen, H. and Verkamo, A. (1994), Efficient algorithms for discovering association rules, *in* 'Papers from the AAAI Workshop on Knowledge Discovery in Databases (KDD'94)', AAAI Press, pp. 181–192.

Meo, R. (2000), 'Theory of dependence values', *ACM Transactions on Database Systems* **25**(3), 380–406.

Morishita, S. and Nakaya, A. (2000), Parallel branch-and-bound graph search for correlated association rules, *in* 'Revised Papers from Large-Scale Parallel Data Mining, Workshop on Large-Scale Parallel KDD Systems, SIGKDD', Vol. 1759 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 127–144.

Morishita, S. and Sese, J. (2000), Transversing itemset lattices with statistical metric pruning, *in* 'Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS'00)', ACM Press, pp. 226–236.

Mozina, M., Demsar, J., Zabkar, J. and Bratko, I. (2006), Why is rule learning optimistic and how to correct it, *in* J. Frnkranz, T. Scheffer and M. Spiliopoulou, eds, 'Proceedings of the 17th European Conference on Machine Learning (ECML'06)', Vol. 4212 of *Lecture Notes in Computer Science*, Springer, pp. 330–340.

Nijssen, S. and Kok, J. (2006), Multi-class correlated pattern mining, *in* 'Proceedings of the 4th International Workshop on Knowledge Discovery in Inductive Databases', Vol. 3933 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 165–187.

Piatetsky-Shapiro, G. (1991), Discovery, analysis, and presentation of strong rules, *in* G. Piatetsky-Shapiro and W. Frawley, eds, 'Knowledge Discovery in Databases', AAAI/MIT Press, pp. 229–248.

Shaffer, J. (1995), 'Multiple hypothesis testing', *Annual Review of Psychology* **46**, 561–584.

Silverstein, C., Brin, S. and Motwani, R. (1998), 'Beyond market baskets: Generalizing association rules to dependence rules', *Data Mining and Knowledge Discovery* **2**(1), 39–68.

Smyth, P. and Goodman, R. (1992), 'An information theoretic approach to rule induction from databases', *IEEE Transactions on Knowledge and Data Engineering* **4**(4), 301–316.

Tan, P.-N., Kumar, V. and Srivastava, J. (2004), 'Selecting the right objective measure for association analysis', *Information Systems* **29**(4), 293–313.

The PLANTS Database (2008). Retrieved 31 December 2008.
URL: *http://plants.usda.gov*

Wang, K., He, Y. and Cheung, D. (2001), Mining confident rules without support requirement, *in* 'Proceedings of the tenth international conference on Information and knowledge management (CIKM1)', ACM, New York, NY, USA, pp. 89–96.

Wang, K., Zhou, S. and He, Y. (2000), Growing decision trees on support-less association rules, *in* 'Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'00)', ACM, New York, NY, USA, pp. 265–269.

Webb, G. (2006), Discovering significant rules, *in* 'Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'06)', ACM Press, New York, USA, pp. 434–443.

Webb, G. I. (2007), 'Discovering significant patterns', *Machine Learning* **68**(1), 1–33.

Yen, S.-J. and Chen, A. (1996), An efficient approach to discovering knowledge from large databases, *in* 'Proceedings of the fourth international conference on on Parallel and distributed information systems (DIS'96)', IEEE Computer Society, Washington, DC, USA, pp. 8–18.

Author Biographies

insert photo

Wlhelmiina Hämäläinen received a M.Th. degree from the University of Helsinki, Finland, in 1998, a M.Sc. degree from the University of Helsinki, Finland, in 2002, and a Ph.Lic. degree from the University of Joensuu, Finland, 2006. She has worked as teacher, lecturer, and researcher both in the University of Helsinki and University of Joensuu since 1996. Currently she is finishing her Ph.D. degree at the Department of Computer Science, University of Helsinki. Her interests include data mining and machine learning algorithms, mathematics, and statistics.

Correspondence and offprint requests to: Wlhelmiina Hämäläinen, Department of Computer Science, University of Helsinki, Finland. Email: whamalai@cs.helsinki.fi