Efficient Statistical Pruning of Association Rules

Alan Ableson[1] and Janice Glasgow[2]

[1] Department of Math and Stats
ableson@mast.queensu.ca
[2] School of Computing
janice@cs.queensu.ca
Queen's University
Kingston, Ontario, Canada

Abstract. Association mining is the comprehensive identification of frequent patterns in discrete tabular data. The result of association mining can be a listing of hundreds to millions of patterns, of which few are likely of interest. In this paper we present a probabilistic metric to filter association rules that can help highlight the important structure in the data. The proposed filtering technique can be combined with maximal association mining algorithms or heuristic association mining algorithms to more efficiently search for interesting association rules with lower support.

1 Introduction

Association mining is the process of identifying frequent patterns in a tabular dataset, usually requiring some *minimum support*, or frequency of the pattern in the data [2]. The discovery of frequent patterns in the data is usually followed by the construction of *association rules*, which portray the patterns as predictive relationships between particular attribute values. Unfortunately, since association mining is an exhaustive approach, it is possible to generate many more patterns than a user can reasonably evaluate. Furthermore, many of these patterns may be redundant. Thus, is it important to develop informed and efficient pruning systems for association mining rules.

A number of methods to filter association mining results have been published, and they can be classified along several lines. First, the filtering can be *objective* or *subjective* [11]. Our goal is to design an objective, or purely computational, filter, both for inter-discipline generality and to avoid the bias introduced by subjective evaluation. In a separate categorization, rule filtering can be done on a rule-by-rule basis [7], or in an incremental manner where rules deemed interesting are gradually added to a rule list set [6] or probability model [12]. We focus our attention on the rule-by-rule approach. This approach is appropriate in application areas where dense data tables lead to the generation of millions of association rules, a set too large to use directly in incremental filtering algorithms. As an added advantage, filtering rules independently allows the straightforward use of batch parallelization of the filtering to produce linear speed-ups.

N. Lavrač et al. (Eds.): PKDD 2003, LNAI 2838, pp. 23–34, 2003.
© Springer-Verlag Berlin Heidelberg 2003

2 Background

It has long been understood in statistics that while not every significant feature of a dataset is interesting, an interesting feature *must* be statistically significant. Non-significant results, by definition, are those features that can be explained as a random effect, and therefore not worthy of further study. For example, an association rule predicting a customer action with 100% accuracy would not be significant if it only covers 2 customers in a large database, and so would not be interesting. This aspect of interestingness is sometimes referred to as the *reliability* of a rule [11].

In many association rule filtering approaches, the measure of significance or reliability has been largely ad-hoc, stemming from Boolean logical theory rather than statistical theory. Boolean approaches typically combine the *support* and *confidence* of a rule in a dual-ordering approach, trying to maximize both support and confidence, with an implied maximization of reliability [5]. Related logical mechanisms for removing redundant association rules use the concept of *closed* itemsets [8,21]. Closure-based methods are most effective in noise-free problems. In these problems, large sets of records contain exactly the same associations, allowing the pruning of redundant subsets of the items without loss of information. In noisier data, fewer rules can be considered redundant according to closure properties, limiting their effectiveness. For such noisy dataset, statistical techniques have been used by other authors, focusing either on pruning association rules [14,19] or identifying correlated attributes involved in association rules [12,17].

3 Problem Statement

In our approach to association rule filtering, we will follow the approach of Liu et al. [14] and use a statistical model to evaluate the reliability of association rules, with a focus on association rules with a single value/item in the consequent of the rule. Furthermore, we pose our problem as one of association rule mining over relational tables, rather than itemsets.

In mining over relational tables, the *input* consists of a table T with a set of N records, $R = \{r_1, \ldots, r_N\}$, and n attributes, $A = \{a_1, \ldots, a_n\}$; all of the attributes take on a discrete set of values, $Dom(a_i)$. We assume the table contains no missing values.

The *output* of an association mining exercise is a set of *patterns*, where a pattern X associates each attribute in a subset of A with a particular value,

$$X = \{< a_{x_1}, v_1 >, < a_{x_2}, v_2 >, ..., < a_{x_k}, v_k >\}$$

such that $v_i \in Dom(a_{x_i})$. A pattern that contains k attribute/value pairs is called a *k-th order* pattern. A record r_i of the table *matches* or *instantiates* a pattern X if, for each attribute a_j in X, r_i contains the value v_j for a_j.

Each of the patterns (also called *itemsets*) has a number of descriptive parameters. We define the *support* of a pattern, $Sup(X|T)$, as the cardinality of the set X_T, denoted $|X_T|$, such that $X_T = \{r_i \in T | r_i$ is an instance of pattern $X\}$. Wherever the table T is the original input table, we simply write $Sup(X)$.

Often patterns are combined to make a larger pattern, through the *composition* of two smaller patterns, X and Y, denoted $X \circ Y$. The composition of two patterns produces a longer pattern including the attribute/value pairs of both input patterns:

$$X \circ Y = \{< a_{x_1}, v_{x_1} >, ..., < a_{x_k}, v_{x_k} >, < a_{y_1}, v_{y_1} >, ..., < a_{y_m}, v_{y_m} >\}$$

To avoid degeneracies, we define composition only for patterns with non-overlapping attributes. Composition of patterns of order k_1 and k_2 results in a pattern of order $k_1 + k_2$.

An *association rule* is a pairing of two patterns, $X \rightarrow Z$, and is interpreted interpreted as a causal or correlational relationship. The *support* and *confidence* of an association rule, denoted $Sup(X \rightarrow Z)$ and $Conf(X \rightarrow Z)$ are defined in terms of support for the patterns and their compositions:

$$Sup(X \rightarrow Z) = Sup(X \circ Z),\ Conf(X \rightarrow Z) = \frac{Sup(X \circ Z)}{Sup(X)}$$

4 Reliable Association Rules

In an association mining study, the set of patterns generated can be straightforwardly turned into a set of association rules [2]. The output of association rule generation will often be a long list of rules, $X_i \rightarrow Z_i$, with the number of rules determined by the table T, the minimum support required for any rule, *minsup*, and possibly a minimum confidence constraint, *minconf*. The two questions we want to address in our filtering, for any particular rule $X \rightarrow Z$, are

1. Is the rule $X \rightarrow Z$ reliable (statistically significant), given $Sup(X)$, $Sup(Z)$, and $Sup(X \circ Z)$?
2. Is the rule $X \rightarrow Z$ an unreliable extension of a lower-order rule, $Y \rightarrow Z$, where $X = Y \circ Q$, for some non-predictive pattern Q?

We approach both questions using a statistical sampling model. Imagine an urn filled with N balls, each ball representing one record of the input table T. For each record, its ball is colored red if it matches the consequent pattern Z, and black if does not. A sample is then taken from the urn of all those balls matching the predictive or antecendent pattern X. We then ask ourselves the question, "is the distribution of red and black balls matching the pattern X significantly different from what we would expect from a random scoop of $Sup(X)$ balls from the urn?" This question can be answered using the hypergeometric probability distribution, which exactly represents the probability of this sampling-without-replacement model. [16]

Specifically, if we have the four values $Sup(X)$, $Sup(Z)$, $Sup(\bar{Z})$, and $Sup(X \circ Z)$ (with shorthand $S_X = Sup(X)$, $S_{X \circ Z} = Sup(X \circ Z)$, etc. for conciseness), the hypergeometric probability distribution, H, is given by:

$$H(n = S_{X \circ Z} | S_Z, S_{\bar{Z}}, S_X) = \frac{C(S_Z, n) C(S_{\bar{Z}}, S_X - n)}{C(S_Z + S_{\bar{Z}}, S_X)} \qquad (1)$$

where $C(N, x) = x!(N - x)!/N!$ is the number of ways to choose x records from a collection of N records.

The function H gives the probability of *exactly* $S_{X \circ Z}$ red balls being selected. To compute a significance value, we need to calculate the *cumulative probability*, or the probability of an outcome as extreme as $S_{X \circ Z}$, where either $P(n \geq S_{X \circ Z})$, or $P(n \leq S_{X \circ Z})$ through the following sums (where n is the number of red balls):

$$P(n \geq S_{X \circ Z} | S_Z, S_{\bar{Z}}, S_X) = \sum_{k=S_{X \circ Z}}^{S_X} H(k | S_Z, S_{\bar{Z}}, S_X) \qquad (2)$$

$$P(n \leq S_{X \circ Z} | S_Z, S_{\bar{Z}}, S_X) = \sum_{k=0}^{S_{X \circ Z}} H(k | S_Z, S_{\bar{Z}}, S_X) \qquad (3)$$

These formulae produce the *p-values* of the rule under the null hypothesis of the hypergeometric sampling model. A low p-value in Equation 2 indicates a higher-than-random confidence for the rule $X \rightarrow Z$, while a low p-value in Equation 3 indicates an interesting *low* confidence rule.

It should be noted that other authors have used p-value ranking approaches, using other statistical measures. The χ^2 measure has been the most popular, but studies have also used the gini index, correlation coefficient, and interest factor (see [19] for a comprehensive presentation of many of these measures). Unfortunately, all of these measures implicitly rely on large sample assumptions, and so their p-value estimates become less reliable as the patterns' support decreases. The hypergeometric distribution, because its explicit counting approach, is an *exact* method, applicable to patterns at all support values. We believe that the hypergeometric is a more appropriate null hypothesis distribution when dealing with association rules that include patterns with low support (less than 50 records). In the rest of the paper, we use the hypergeometric distribution exclusively, keeping in mind that other distributions could be used.

5 Filtering Algorithms

The hypergeometric probability distribution, or any other appropriate null hypothesis distribution such as χ^2, can be used to test the significance of association rules in two ways. The simplest way is to evaluate the reliability of every association rule discovered, $X \rightarrow Z$ by computing its p-value (using Equations 3 and 2) with the p-values being computed relative to the baseline distribution of the consequent pattern, Z. In particular, for any rule, $X_i \rightarrow Z_i$ found during association mining, we can define the two-tailed p-value of rule, p_i, as follows:

$$p_i = min \begin{pmatrix} P(n \geq S_{X_i \circ Z_i} | S_Z, S_{\bar{Z}}, S_X) \\ P(n \leq S_{X_i \circ Z_i} | S_Z, S_{\bar{Z}}, S_X) \end{pmatrix}$$

If p_i is less than some user-defined p-value threshold p_{max}, we consider the rule $X_i \rightarrow Z_i$ to be interesting.

This p-value ranking is simple, and is computationally linear in the number of association rules found. It provides a straightforward statistical approach to rank and filter association rules. The disadvantages are that many similar rules will be evaluated, and, if a particular rule is significant, e.g. $X \to Z$, then adding spurious extensions to the antecedent is also likely to produce a significant rule e.g. $(X \circ Y) \to Z$, even if Y is a pattern which has no predictive relationship with Z.

To overcome the problems of spurious rule extensions, a statistical model may be used in a incremental way to gradually prune away non-interesting parts of an association rule, leaving only the interesting part. This process is laid out in Algorithm 1.

Algorithm 1 *Incremental Probabilistic Pruning* Input*: A table (T), a minimum support (minsup), and a p-value cutoff (p_{max}).* Output*: a set of association rules, R_{out}.*

1. *Create an empty set of association rules, $R_{out} = \phi$.*
2. *Construct a set of association rules, $I = \{X_i \to Z_i\}$ from the set of records in T, constraining the search with the minimum support, minsup.*
3. *For each association rule $X_i \to Z_i$, (called $X \to Z$ for clarity in this step),*
   (a) *Refer to T to find the supports related to the association rule, $S_{X \to Z}$,*
   (b) *Compute the two-tailed p-value of the association rule given every possible sub-rule $Y_j \to Z$, where $X = Y_j \circ Q_j$, where Q_j is a first-order pattern, and Y_j is one order less than the pattern X. Take the maximum of the p-values over the possible patterns Y_j:*

$$p_{ij} = min \begin{pmatrix} P(n \geq S_{X \circ Z} | S_{Y_j \circ Z}, S_{Y_j \circ \bar{Z}}, S_X) \\ P(n \leq S_{X \circ Z} | S_{Y_j \circ Z}, S_{Y_j \circ \bar{Z}}, S_X) \end{pmatrix}$$

$$p_i = \underset{Y_j}{max} \ (p_{ij})$$

   using Equations 2 and 3 to compute the hypergeometric probabilities.
   (c) *If $p_i < p_{max}$, then none of the sub-rules $Y_{ij} \to Z_i$ can explain the rule $X_i \to Z_i$, so add the rule $X_i \to Z_i$ to the collection R_{out}.*
   (d) *If $p_i \geq p_{max}$, find the sub-rule $Y_{ij} \to Z_i$ for which the p_{ij} was maximal, and return to step 3, with $Y_{ij} \to Z_i$ instead of $X_i \to Z_i$.*

6 Implications of Pruning Algorithm for Association Mining

The idea of using subsuming patterns to define the relative interestingness of association rules was proposed earlier by Liu et. al. [14], using the χ^2 distribution rather than hypergeometric to measure rule significance. They performed their pruning following the incremental Apriori approach [1]. This involved finding and pruning all first-order rules, then second-order rules, etc. In dense data tables, as found by Liu et al., there are many association rules with relatively high support (above 20%), but few rules that pass by significance pruning as outlines in Algorithm 1. However, since the rules with high support are most likely to be those already known in the domain, it behooves us to search for interesting rules of lower support, which is hopelessly infeasible using a pure Apriori-based approach.

Fortunately, other association mining algorithms exist which do not require the explicit construction of every possible sub-pattern for an association rule. There are several algorithms that efficiently search for *maximal frequent patterns* [13,9]. A frequent pattern X is *maximal* if has no superset that is frequent: $Sup(X) \geq minsup$, and $Sup(X \circ Y) < minsup$ for any pattern Y. Two examples of algorithms that identify maximal itemsets are Max-Miner [13] and GenMax [9]. There are also heuristic association mining algorithms, such as SLAM [18] which do not use the recursive search strategy of the previous algorithms. Our hypothesis is that using the incremental pruning procedure in Algorithm 1, we should identify the roughly the same set of interesting rules whether we start from a relatively small set of maximal frequent patterns (e.g. GenMax), the complete set of frequent patterns (e.g. Apriori), or a heuristically-generated subset of patterns (e.g. SLAM).

7 Applications

There were two particular questions we wanted to answer in evaluating our association rule pruning. The first was what percentage of rules identified on different datasets could be safely be pruned away using Algorithm 1. The second question was how effective incomplete searches for frequent patterns followed by pruning would be compared to more exhaustive searches followed by pruning. Incomplete association mining algorithms can be orders of magnitude faster than exhaustive algorithms, and produce orders of magnitude fewer associations. If the pruning leads to roughly the same results using these different association mining tools, then there is a considerable computational advantage to be gained by pruning only the rules generated by the incomplete association mining algorithms.

In all the experiments, we computed p-values using the hypergeometric probability calculation described in [20]. We found the algorithm sufficiently fast for the number of records encountered, and it provided exact estimates of the probabilities for rules with small support. If faster computations are necessary, the hypergeometric distribution can be approximated by an appropriate binomial, normal or χ^2 distribution.

7.1 Description of Datasets

To evaluate the pruning technique, one simulated collection and two real-world datasets were considered.

The simulated datasets were designed to test the sensitivity and specificity of the pruning approach in Algorithm 1. Tables were created with 51 binary-valued attributes (values “0” and “1”) and 1,000 records. The first 50 attributes were treated as the predictor attributes, and the 51st attribute was the *outcome attribute*, or the consequent attribute for association rules. The goal was to find patterns predictive of the positive value, “1”, of this 51st attribute. The 50 predictor attributes were generated independently, using a uniform probability for each binary value. With this distribution, all predictive patterns of size n have an expected support of $1,000/2^n$.

The distribution for the outcome attribute was defined using two parameters, n_p and o_p, where n_p specified the size of the predictive pattern, and o_p the relative odds of a positive outcome if a record was an instance of the predictive pattern. For ease of identification, a pattern of size n_p was considered to be the pattern of n_p 1’s on the first n_p attributes. For example, if we chose $n_p = 3$ and $o_p = 4$, then records being an instance of the pattern “1,1,1” as the values for the first three attributes would have a 4 times greater chance of having a 1 in the outcome than the other records. Records not matching the predictive pattern had a probability of 1/2 of being an outcome of 1.

The rationale behind this simulation is that it generates dense datasets with large numbers of association rules, of which only a single rule is actually predictive. By controlling the size of the predictive rule by n_p, we are effectively setting the support to $1000/2^{n_p}$, and by setting the odds we can control the interestingness of the rule.

The real datasets we studied have been used previously in evaluations of algorithms for maximal frequent pattern identification. For clarity, we focused our attention on the *chess*, and *mushroom* datasets, (available from the UCI Machine Learning Repository [4]). The chess dataset contains 3,196 records with 23 attributes, while the mushroom dataset contains 8,124 records of 23 attributes.

7.2 Association Mining Algorithms

For each of the datasets, we performed searches for patterns associated with an outcome of interest. In the simulated datasets, we searched for patterns with a ‘1’ in the fifty-first attribute; for the chess dataset, we searched for patterns that contained the outcome “won”, and for the mushroom dataset, we searched for patterns that contained the outcome “edible”.

We performed the search for these predictive patterns using four different association mining algorithms. Our goal was to decide whether the pruning technique required the explicit listing of all associations or whether more efficient but less exhaustive association mining searches could provide the same results after pruning. The four algorithms we used were Apriori [2], FPMine* [10], SLAM [18], and MAX.

Apriori was used in its normal fashion to find a complete set of patterns, given a particular minimum support constraint. We modified the FPMine algorithm of [10] slightly so it would not output every possible subset of patterns found when it had uncovered a path-tree in its recursion step; we called our variation FPMine*. With this change, FPMine* outputs fewer associations at a given minimum support than Apriori, but the set is still generally large and will include the maximal frequent patterns. SLAM is a heuristic association

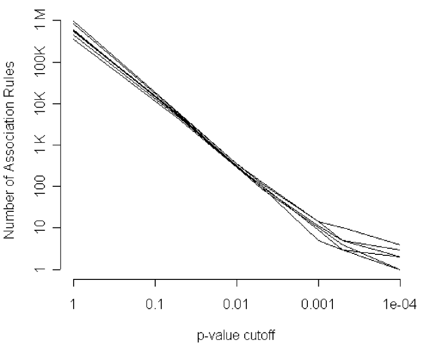

Fig. 1. Log-log plot of the number of association rules reported at different p-value cutoffs for Algorithm 1, applied to the simulated data. Different lines indicate different parameter settings, n_p and o_p, of the simulator for data generation.

mining algorithm, which tends to generate fewer associations than Apriori and FPMine*, but can use a lower *minsup*. MAX was written as a simple recursive search for maximal patterns (see [13] and [9] for more efficient algorithms for mining maximal patterns).

8 Results

We present the results for the filtering of association rules below, discussing first the results using simulated datasets and then using the real datasets.

8.1 Simulated Datasets

For identifying rules on the simulated datasets, we only used the Apriori algorithm, with a minimum support cutoff of $minsup = 40$ records out of 1,000. With this support, Apriori generated between 500,000 and 1,000,000 association rules, of which only a single rule was actually predictive. Since Algorithm 1 takes a p-value cutoff as a parameter, we first studied the number of unique association rules left using different p-value cutoffs. The graph in Figure 1 shows the steep decline in the number of interesting rules as the p-value threshold is lowered[1].

From the graph in Figure 1, it is clear that the pruning dramatically reduced the number of patterns reported, so few non-predictive patterns are reported when the p-value threshold is set high enough. Our next concern was the sensitivity of the test: how frequently was the real pattern reported after pruning? We considered pattern sizes (n_p) from {1,2,3,4}, and predictive odds (o_p) from

[1] It should be noted that the pruning rule is performing multiple statistical tests for each of the many association rules, and so a “typical” p-value cutoff like 0.01 is not appropriate: lower (more significant) values should be used to obtain more realistic significance estimates. A discussion of multiple testing is presented in [3].

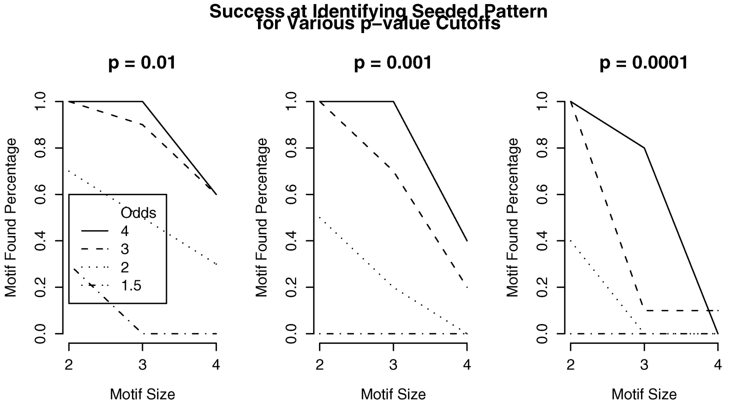

Fig. 2. For the simulated data, using different p-values, the seeded motif is reported less often if it is relatively infrequent (larger motif sizes) or less predictive (lower odds).

{1.5, 2, 3, 4}. For each possible pair (n_p, o_p), we generated ten tables using the simulator distribution, performed a search for frequent patterns (*minsup* of 40), and pruned the resulting set using Algorithm 1.

In Figure 2, we see the percentage of the runs for which the correct motif was found after pruning, for different p-values of the pruning algorithm. As expected, when the motifs were shorter and had higher odds, they were more significant and more consistently reported. Motifs with odds of 1.5 were only reported occasionally when the motif size was short (size 2), and never otherwise, even for least restrictive p-value of 0.01. This is unsurprising, given the low odds and relatively low prevalence of the rule; the occasional significant finding is the result of sampling variation.

Motifs of size 3 or 4 with odds of 1.5 have too low a support to ever be significant even at $p = 0.01$ levels.

The results of the association mining on simulated data indicate that the approach is effective at removing the vast majority of statistically non-interesting patterns, while still identifying real motifs if they achieve the specified statistical confidence level. We now consider the application of the pruning algorithm in the context of real data.

8.2 Real Datasets

For the real datasets, chess and mushroom, we used all four association mining algorithms, Apriori, FPMine*, SLAM and MAX. For Apriori and FPMine*, we experimented to find minimum support values that produced roughly 2 million patterns associated with the outcome of interest (winning the chess game, edible mushrooms). Since FPMine* does not report all patterns above the minimum support threshold, it could use a lower minimum support, and identify less frequent patterns. We ran MAX at various levels of support for which run times were reasonable (less than an hour). SLAM, being an iterative procedure, can

Table 1. Number of patterns found before and after pruning for different association mining algorithms, using a pruning p-value of 0.0001.

	Dataset		
	Chess		
	minsup	Patterns	Pruned Patterns
Apriori	860	2195736	1467
FPMine*	760	2078291	2186
SLAM	300	243782	3541
MAX	300	20356	568
	Mushroom		
	minsup	Patterns	Pruned Patterns
Apriori	280	2334446	3182
FPMine*	70	2337408	5122
SLAM	40	26950	1563
MAX	40	4859	231

be run for any number of iterations, regardless of the minimum support. We selected a run-time of 10 minutes on each dataset to search for patterns (similar to Apriori and FPMine*). For ease of comparison, SLAM used a minimum support equal to that used with MAX.

After each association mining algorithm was used on both datasets, the patterns found were pruned using Algorithm 1. The number of patterns before and after the pruning are given in Table 1.

In Table 1, we see the clear monotonicity in the minimum supports used for the different algorithms. Furthermore, the incomplete searches (SLAM and MAX) generate orders of magnitude fewer associations than the more complete searches (Apriori and FPMine*). When different p-value cutoffs were used in the pruning, the relative number of pruned motifs for each algorithm stayed roughly constant (not shown).

We also studied the relative overlap of the patterns reported by each method. We used the patterns generated by Apriori as the baseline for comparison. In general, using the FPMine* algorithm to find patterns with lower support resulted in a duplication of the patterns found by Apriori, with the addition of new significant patterns with lower support. For the chess dataset, all of the pruned patterns from Apriori were found in the pruned FPMine* patterns, with FPMine* having an additional 719 novel patterns. For the mushroom dataset, there were some patterns found in the pruned Apriori results (169 patterns), while FPMine* reported an additional 2109 patterns beyond the set found by Apriori. This indicated that overall, if more significant patterns were desired, then using an approach like FPMine*, with a relatively low minimum support, would be more rewarding than performing a complete search with higher support using Apriori.

The patterns found while pruning the SLAM and MAX results were quite different from those found by the more complete search methods. In the chess dataset, of MAX's pruned patterns, roughly two-thirds were not present in the patterns from Apriori, but 1123 patterns from Apriori were not found by MAX. In this situation, there is argument for using several search algorithms to maximize the likelihood of finding a comprehensive set of interesting patterns.

As we hoped, the heuristic association mining algorithm SLAM provided a complementary approach between the extremes of discovering all patterns and only the maximal patterns. It generated far fewer patterns than the complete methods, of which a much higher proportion pruned down to unique significant patterns.

9 Discussion

We have shown that the proposed filtering algorithm can be used to reliably detect truly predictive patterns in data tables with many spurious associations. We have further shown that the algorithm can be used to complement a variety of association mining algorithms, from complete searches to maximal assocation mining. Regardless of the search algorithm, the number of significant associations is always far smaller than the raw set returned, indicating the usefulness of statistical pruning as a post-processing step for any association mining. Furthermore, by selecting different p-value cutoffs for the pruning, the size of the final set of patterns can be easily controlled.

There are several desirable features about this filtering algorithm. Firstly, the filtering can be applied to every predictive pattern independently of the others, meaning that batch parallelism could be used to simultaneously prune a large number of patterns. The parallel overhead would the re-combination of the relatively small number of pruned patterns into a single set of unique patterns.

Also, the proposed pruning method can be used to evaluate the statistical reliability of rules against a more general probability model. In our examples, we evaluated the support of various patterns by counting records in the dataset. However, a more general probability model could be used to compute the expected support levels, such as the maximum entropy model described in [15].

10 Conclusion

We have presented a statistical approach for filtering and pruning predictive patterns identified through association mining. We have further shown that this filtering can be used with a variety of association mining algorithms, allowing a progressive filtering of a large collection of predictive patterns down to a relatively small set of significant patterns. This pruning can be performed in isolation, or as a pre-processing step for more computationally expensive pattern filtering algorithms.

Acknowledgments

We would like to thank Molecular Mining Corporation for permission to use SLAM algorithm, and the Natural Science and Engineering Research Council of Canada and the Institute for Robotics and Intelligent System for financial support for the research. We would also like to thank the three anonymous referees for their helpful comments and for identifying important previous work.

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499, 12–15 1994.
2. R. Agrawal and T. Imielinski A. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Intl. Conference on Management of Data*, pages 207–216, 26–28 1993.
3. Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, 57(1):389–300, 1995.
4. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
5. D. Shah et al. Interestingness and pruning of mined patterns. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 1999.
6. H. Toivonen et al. Pruning and grouping of discovered association rules, 1995.
7. M. Klemettinen et al. Finding interesting rules from large sets of discovered association rules. In *CIKM 1994)*, pages 401–407, 1994.
8. Y Bastide et al. Mining minimal non-redundant association rules using frequent closed itemsets. *Lecture Notes in Computer Science*, 1861:972, 2000.
9. K. Gouda and M.J. Zaki. Efficiently mining maximal frequent itemsets. In *ICDM*, pages 163–170, 2001.
10. J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *ACM SIGMOD Intl. Conference on Management of Data*, pages 1–12, 05 2000.
11. F. Hussain, H. Liu, and H. Lu. Relative measure for mining interesting rules. In *The Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2000.
12. S. Jaroszewicz and D. A. Simovici. Pruning redundant association rules using maximum entropy principle. In *Advances in Knowledge Discovery and Data Mining, PAKDD*, pages 135–147, 2002.
13. R. J. Bayardo Jr. Efficiently mining long patterns from databases. In *ACM SIGMOD Intl. Conference on Management of Data*, pages 85–93, 1998.
14. Bing Liu, Wynne Hsu, and Yiming Ma. Pruning and summarizing the discovered associations. In *Knowledge Discovery and Data Mining*, pages 125–134, 1999.
15. Dmitry Pavlov, Heikki Mannila, and Padhraic Smyth. Beyond independence: Probabilistic models for query approximation on binary transaction data. Technical Report UCI-ICS TR-01-09, UC Irvine, 2001.
16. S. M. Ross. *Introduction to Probability Models.* Academic Press, 1972.
17. C. Silverstein S. Brin, R. Motwani. Beyond market baskets: Generalizing association rules to correlations. In *SIGMOD Conference*, pages 265–276, 1997.
18. E. Steeg, D. A. Robinson, and E. Willis. Coincidence detection: A fast method for discovering higher-order correlations in multidimensional data. In *KDD 1998*, pages 112–120, 1998.
19. P. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *ACM SIGKDD*, 2002.
20. T. Wu. An accurate computation of the hypergeometric distribution function. *ACM Transactions on Mathematical Software (TOMS)*, 19(1):33–43, 1993.
21. M. Zaki. Generating non-redundant association rules. In *KDD 2000*, pages 34–43, 2000.