Mining Non-Derivable Association Rules

Bart Goethals*   Juho Muhonen   Hannu Toivonen
Helsinki Institute for Information Technology
Department of Computer Science
University of Helsinki
Finland

Abstract

Association rule mining typically results in large amounts of redundant rules. We introduce efficient methods for deriving tight bounds for confidences of association rules, given their subrules. If the lower and upper bounds of a rule coincide, the confidence is uniquely determined by the subrules and the rule can be pruned as redundant, or *derivable*, without any loss of information. Experiments on real, dense benchmark data sets show that, depending on the case, up to 99–99.99% of rules are derivable. A lossy pruning strategy, where those rules are removed for which the width of the bounded confidence interval is 1 percentage point, reduced the number of rules by a furher order of magnitude. The novelty of our work is twofold. First, it gives absolute bounds for the confidence instead of relying on point estimates or heuristics. Second, no specific inference system is assumed for computing the bounds; instead, the bounds follow from the definition of association rules. Our experimental results demonstrate that the bounds are usually narrow and the approach has great practical significance, also in comparison to recent related approaches.

1 Introduction

Association rule mining often results in a huge amount of rules. Attempts to reduce the size of the result for easier inspection can be roughly divided to two categories. (1) In the subjective approaches, the user is offered some tools to specify which rules are potentially interesting and which are not, such as templates [KMR+94] and constraints [NLHP98, GVdB00]. (2) In the objective approaches, user-independent quality measures are applied on association rules. While interestingness is user-dependent to a large extent, objective measures are needed to reduce the redundancy inherent in a collection of rules.

The objective approaches can be further categorized by whether they measure each rule independently of other rules (e.g., using support, confidence, or lift) or address rule redundancy in the presence of other rules (e.g., being a rule with the most general condition and the most specific con-

sequent among those having certain support and confidence values). Obviously only approaches of the latter type can potentially address redundancy between rules. Our work will be in this category.

We show how the confidence of a rule can be bounded given only its subrules (the condition and consequent of a subrule are subsets of the condition and consequent of the superrule, respectively). It turns out, in practice, that the lower and upper bounds coincide often, and thus the confidence can be derived exactly. We call these rules derivable: they can be considered redundant and pruned without loss of information. We also consider lossy pruning strategies: a rule is pruned if the confidence can be derived with a high accuracy, i.e., if the bounded interval is narrow.

Unlike practically all previous work on pruning association rules by their redundancy, our method for testing the redundancy of a rule is based on deriving absolute bounds on its confidence rather than using an ad hoc estimate. Given an error bound, we can thus guarantee that the confidence of the pruned rules can be estimated (derived) within the bounds. No (arbitrary) selection of a derivation method is involved: the bounds follow directly from the definitions of support and confidence. (A pragmatic choice we will make is that only subrules are used to derive the bounds; see below.)

In a sense, the proposed method is a generalization of the idea of only outputting the free or closed sets [PBTL99, BBR00]. Using free sets and closed sets corresponds, however, to only pruning out rules for which we know the confidence is one. In the method we propose, the confidence can have any value, and the rule is pruned if we can derive that value. Closed sets and related pruning techniques actually work on sets, not on association rules. There are other, more powerful pruning methods for sets. In particular, our work is an extension of the work on non-derivable sets [CG02] to non-derivable association rules. The method is simple, yet it has been overlooked by previous work on the topic.

Optimally, the final collection of rules should be understandable to the user. The minimal collection of rules from which all (pruned) rules can be derived would have a small

---

*Current affiliation: Dept. of Math and Computer Science, University of Antwerp, Belgium

size, but it would most likely be difficult for the user to see why the rest of the rules were pruned and what their confidences must be. We consider different alternatives, including the relatively popular compromise of grouping rules by their consequents and ordering them by the size of the condition. Then, each rule is checked for redundancy given only its subrules having the exactly same consequent, and only non-derivable rules are output.

As a summary, our contributions are the following. We give theoretically sound methods for bounding the confidence of an association rule given its subrules. We then propose to prune as redundant those association rules for which the confidence can be derived exactly or within a guaranteed, user-specified error bound. Experiments with several real data sets (chess, connect, mushroom, pumsb) demonstrate great practical significance: 99–99.99% of rules had (exactly) derivable confidences. Further significant pruning is obtained by removing rules derivable within just ± 0.5 percentage points: the remaining number of rules was only 0.005%–0.04%.

The rest of this article is organized as follows. Section 2 reviews the basic concepts and related work. In Section 3 we define non-derivable association rules and give methods for deriving absolute and tight upper and lower bounds for rule confidences. In Section 4 we give experimental results on a number of real data sets. Section 5 contains our conclusions.

2 Problem Definition and Related Work

The association rule mining problem can be described as follows [AIS93]. We are given a set of items $\mathcal{I}$ and a database $\mathcal{D}$ of subsets of $\mathcal{I}$ called transactions. An association rule is an expression of the form $X \Rightarrow Y$, where X and Y are sets of items, X is called the condition, and Y the consequent. The *support* of a set I is the number of transactions that include I. A set is called *frequent* if its support is no less than a given minimal support threshold. An association rule is called frequent if $X \cup Y$ is frequent and it is called *confident* if the support of $X \cup Y$ divided by the support of X exceeds a given minimal confidence threshold. The goal is now to find all association rules over $\mathcal{D}$ that are frequent and confident.

Typically, for reasonable thresholds, the number of association rules can reach impractical amounts, such that analyzing the rules themselves becomes a challenging task. Moreover, many of these rules have no value to the user since they can be considered redundant. Removing these redundant rules is an important task which we tackle in this paper.

Previous work on pruning redundant association rules is typically based on a decision rule that compares the confidence or support of an association rule to similar rules. For instance, rule $X \Rightarrow Y$ is a "minimal non-redundant association rule" [BPT+00] if there is no rule $X' \Rightarrow Y'$ with $X' \subset X, Y' \supset Y$ such that $supp(XY) = supp(X'Y')$

and $conf(X \Rightarrow Y) = conf(X' \Rightarrow Y')$. A similar but not identical definition is given for "closed rules" in [Zak00] or "minimal rules" in [ZP03]. A recent proposal is that rule $X \Rightarrow Y$ is not a "basic association rule" [LH04] if there exists $X' \subset X$ such that for all $X'', X' \subseteq X'' \subseteq X$, $conf(X \Rightarrow Y) = conf(X'' \Rightarrow Y)$. Our proposal differs from these techniques in two significant aspects. First, it has a wider applicability: the above-mentioned concepts only apply for rules with exactly the same confidence. Second, these techniques use specific inference systems to decide when a rule is pruned, and in order to know the confidence or support of a pruned rule, the user must use the exact same inference system. In our proposed technique, the bounds follow from the definition of association rules.

Another approach is to estimate rule confidence from a collection of other rules. For example, the maximum entropy technique declares a rule to be redundant if its true confidence is close to the estimate [MPS99, JS02]. In theory, the maximum entropy principle yields consistent estimates in the sense that the value is possible, i.e., it is within the bounds implied by the constraints used. There are some critical issues in its application to rule pruning, however. First, the principle does not give any guarantees for the error bounds. Second, a pruning strategy based on removing rules for which the error is below a given upper bound alleviates the first issue, but at the cost of assuming maximum entropy principle as the inference system. Finally, it is computationally demanding to compute the maximum entropy solution. Practical alternatives rely on approximations, and then lose the benefit of producing consistent estimates.

For a good and quite recent, yet brief overview of attempts to find non-redundant association rules, see reference [LH04].

Some of the approaches mentioned above [BPT+00, Zak00] utilize the concept of *closed sets*. A set is called closed if it has no proper superset with the same support; from this, it follows that a non-closed set X implies the rest of its closure with 100% certainty, i.e., the confidence of rule $X \Rightarrow Y$ equals 1 when Y is a subset of X's closure. Given a non-closed set X, any set Y in its closure, and a rule $X \Rightarrow Z$, it has been proposed to prune rules of the form $XY \Rightarrow Z$ and $X \Rightarrow YZ$ as redundant since their frequencies and confidences are identical with the rule $X \Rightarrow Z$. As mentioned above, this approach makes assumptions, and without knowing them the user cannot know why rule $XY \Rightarrow Z$ was pruned.

A good amount of work has focused on finding condensed representations for frequent sets by pruning redundant sets. Obviously, the number of association rules is even much larger and hence the problem is even more important to solve. In the case of frequent sets, the most successful condensed representation is the notion of closed sets: all frequent sets can be derived from the closed frequent sets (or

frequent generators). δ-free sets generalize this notion to "almost closed" sets [BBR00].

More recently, a more powerful method for pruning frequent sets has been presented, called *non-derivable sets* [CG02]. The main idea is to derive a lower and an upper bound on the support of a set, given the supports of all its subsets. When these bounds are equal (the support of) the set is *derivable*. In this paper, we extend this work in a natural way to association rules: we introduce similar derivation techniques to find tight bounds on the confidence of a rule, given its subrules.

The problem we attack can be formulated as follows. Given the set $\mathcal{R}$ of association rules (with respect to a given frequency threshold, confidence threshold, and database $\mathcal{D}$), choose a subset $\mathcal{R}' \subset \mathcal{R}$ such that the confidence of every pruned rule $R \in \mathcal{R}' \setminus \mathcal{R}$ can be derived up to a user-specified error limit, possibly zero, from its subrules. Rule $X' \Rightarrow Y'$ is a *subrule* of $X \Rightarrow Y$ iff $X' \subseteq X$ and $Y' \subseteq Y$; selecting only the subrules to derive the confidence of a given rule should improve the understandability of the results. (In this paper, the term subrule will refer to proper subrules, i.e., subrules not equal to the original rule.) In other words, rule $X \Rightarrow Y$ is *derivable* and redundant, if its confidence can be derived from the confidences and supports of its subrules; otherwise it is *non-derivable*. Note that being derivable is a function of the subrules: the actual rule confidence and support are not needed for knowing whether the rule is derivable.

Before going to the methods, we would like to remind the readers that obviously redundancy is not the only reason why some association rules are uninteresting. Interestingness is often subjective, and tools such as templates or other syntactical constraints can be very useful. Subjective interestingness is, however, outside the scope of this paper.

3 Non-Derivable Association Rules

We now show how to derive lower and upper bounds for the confidence of an association rule, given its subrules. We start by reviewing the technique to derive bounds on the support of a set [CG02].

3.1 Sets The main principle behind the support derivation technique used for mining non-derivable sets is the inclusion-exclusion principle [GS00]. For any subset $J \subseteq I$, we obtain a lower or an upper bound on the support of I using one of the following formulas.

If $|I \setminus J|$ is odd, then

$$(3.1) \qquad supp(I) \leq \sum_{J \subseteq X \subset I} (-1)^{|I \setminus X|+1} supp(X).$$

If $|I \setminus J|$ is even, then

$$(3.2) \qquad supp(I) \geq \sum_{J \subseteq X \subset I} (-1)^{|I \setminus X|+1} supp(X).$$

For example, in Figure 1, we show all possible rules to derive the bounds for a given set $\{abcd\}$.

When the smallest upper bound equals the highest lower bound, then we have actually obtained the exact support of the set solely based on the supports of its subsets. These sets are called *derivable*, and all other sets *non-derivable*. The collection of non-derivable sets has several nice properties.

PROPERTY 3.1. *[CG02] The size of the largest non-derivable set is at most $1 + \log|D|$ where $|D|$ denotes the total number of transactions in the database.*

PROPERTY 3.2. *[CG02] The collection of non-derivable sets is downward closed. In other words, all supersets of a derivable set are derivable, and all subsets of a non-derivable set are non-derivable.*

A less desirable property is that the number of bounds for a given itemset is exponential in the size of the itemset. For more results and discussions, we refer the interested reader to [CG02].

3.2 Association Rules Now, consider a rule $X \Rightarrow Y$ and assume all its (proper) subrules are known, i.e., their supports and confidences are given and hence, also the support of all proper subsets of $X \cup Y$. In order to compute bounds for the confidence of that rule, we bound the support of $X \cup Y$ using the above described technique and divide the lower and upper bound by the support of X, resulting in a lower and upper bound for the confidence of $X \Rightarrow Y$. The goal is to find and remove all *derivable* association rules, i.e., rules for which the lower and the upper bounds of confidence are equal. From this procedure, the following property is readily verified.

PROPERTY 3.3. *Given all (proper) subrules of association rule $X \Rightarrow Y$: $X \Rightarrow Y$ is derivable if and only if $X \cup Y$ is a derivable set.*

This leads to an association rule pruning method which can be represented as a simple modification to the original association rule generation algorithm in which only non-derivable itemsets are used.

Note that when considered as sets in separation, X can be a non-derivable itemset while the set $X \cup Y$ is a derivable itemset, cfr. Property 3.2. A straightforward application of non-derivability of itemsets to association rule mining would be to output rules in which the condition X is non-derivable (regardless of whether the union $X \cup Y$ is).

We next consider some interesting, more restricted cases of pruning. When considering the possible redundancy of a specific association rule, it is probably natural and easier to focus only on those rules which have exactly the same condition or exactly the same consequent. Such a compromise results in less pruning but is likely to increase the understandability of pruning.

$$\begin{aligned}
supp(abcd) &\geq supp(abc) + supp(abd) + supp(acd) + supp(bcd) - supp(ab) - supp(ac) - supp(ad) \\
&\quad - supp(bc) - supp(bd) - supp(cd) + supp(a) + supp(b) + supp(c) + supp(d) - supp(\{\}) \\
supp(abcd) &\leq supp(a) - supp(ab) - supp(ac) - supp(ad) + supp(abc) + supp(abd) + supp(acd) \\
supp(abcd) &\leq supp(b) - supp(ab) - supp(bc) - supp(bd) + supp(abc) + supp(abd) + supp(bcd) \\
supp(abcd) &\leq supp(c) - supp(ac) - supp(bc) - supp(cd) + supp(abc) + supp(acd) + supp(bcd) \\
supp(abcd) &\leq supp(d) - supp(ad) - supp(bd) - supp(cd) + supp(abd) + supp(acd) + supp(bcd) \\
supp(abcd) &\geq supp(abc) + supp(abd) - supp(ab) \\
supp(abcd) &\geq supp(abc) + supp(acd) - supp(ac) \\
supp(abcd) &\geq supp(abd) + supp(acd) - supp(ad) \\
supp(abcd) &\geq supp(abc) + supp(bcd) - supp(bc) \\
supp(abcd) &\geq supp(abd) + supp(bcd) - supp(bd) \\
supp(abcd) &\geq supp(acd) + supp(bcd) - supp(cd) \\
supp(abcd) &\leq supp(abc) \\
supp(abcd) &\leq supp(abd) \\
supp(abcd) &\leq supp(acd) \\
supp(abcd) &\leq supp(bcd) \\
supp(abcd) &\geq 0
\end{aligned}$$

Figure 1: Bounds on $supp(abcd)$.

3.3 Fixed Consequent First we consider the case of a fixed consequent. In other words, the derivability (redundancy) of a rule is a function of those subrules that explain the same consequent. We handle this case as two separate subclasses of rules, those with a single item consequent and those with multiple items in the consequent.

First consider rules $X \Rightarrow Y$ with $|Y| = 1$. Given all its subrules with the same consequent and their respective supports and confidences, we immediately obtain the supports of all subsets of $X \cup Y$, except of the sets X and $X \cup Y$ themselves.

EXAMPLE 1. *Consider the rule $abc \Rightarrow d$. From each of its subrules, e.g., $ab \Rightarrow d$, we obtain the support of two subsets of $abcd$: the support of abd (the support of the rule) and the support of ab (the support of the rule divided by its confidence).*

rule	sets
$ab \Rightarrow d$	ab, abd
$ac \Rightarrow d$	ac, acd
$bc \Rightarrow d$	bc, bcd
$a \Rightarrow d$	a, ad
$b \Rightarrow d$	b, bd
$c \Rightarrow d$	c, cd
$\{\} \Rightarrow d$	$\{\}, d$

The only two subsets of $abcd$ that are missing are abc and $abcd$, i.e., exactly those needed to compute the confidence of the desired rule.

Thus, given the subrules of $X \Rightarrow Y$ with the same consequent, the support of X can be directly bounded.

For bounding the support of $X \cup Y$, however, information about X is missing, and we cannot simply use all derivation formulas. To solve this, we first compute the bounds for X, and then we compute the bounds for $X \cup Y$ for every possible value of X. As a result, we have a set of triples (v, l, u) with v a possible support value for X and l and u the corresponding lower and upper bound for $X \cup Y$ respectively.

EXAMPLE 2. *Suppose we want to bound the confidence of the rule $ab \Rightarrow c$, given the following supports.*

$supp(ac)$	$=$	3
$supp(bc)$	$=$	3
$supp(a)$	$=$	7
$supp(b)$	$=$	7
$supp(c)$	$=$	5
$supp(\{\})$	$=$	10

Then, bounding ab results in a lower bound of $4 = 7 + 7 - 10 = supp(a) + supp(b) - supp(\{\})$, and an upper bound of $7 = supp(a) = supp(b)$. Then for every possible value of the support of ab, we compute the bounds for the support of abc and the corresponding bounds for the confidence of $ab \Rightarrow c$.

	$supp(abc)$	$conf(ab \Rightarrow c)$
$supp(ab) = 4$	$[1, 1]$	$[1/4, 1/4]$
$supp(ab) = 5$	$[1, 2]$	$[1/5, 2/5]$
$supp(ab) = 6$	$[2, 3]$	$[2/6, 3/6]$
$supp(bb) = 7$	$[3, 3]$	$[3/7, 3/7]$

Hence, we can conclude that the confidence interval of $ab \Rightarrow c$ is $[1/5, 1/2]$.

As the example above shows, it is not sufficient to use only values at the lower and the upper bounds of X when computing the bounds for $X \cup Y$: the extreme values for the confidence may occur at intermediate possible values of X.

Also note that a rule $X \Rightarrow Y$ can be derivable even if X is not. This is the case when all the bounds of $X \cup Y$, for every possible value of X, result in the same equal upper and lower bound on the confidende of $X \Rightarrow Y$, as illustrated in the following example.

EXAMPLE 3. *Suppose we want to bound the confidence of the rule $ab \Rightarrow c$, given the following supports.*

$supp(ac)$	$=$	7
$supp(bc)$	$=$	7
$supp(a)$	$=$	7
$supp(b)$	$=$	7
$supp(c)$	$=$	10
$supp(\{\})$	$=$	10

Then, bounding ab results in a lower bound of $4 = 7 + 7 - 10 = supp(a) + supp(b) - supp(\{\})$, and an upper bound of $7 = supp(a) = supp(b)$. Then for every possible value of the support of ab, we compute the bounds for the support of abc and the corresponding bounds for the confidence of $ab \Rightarrow c$.

	$supp(abc)$	$conf(ab \Rightarrow c)$
$supp(ab) = 4$	$[4, 4]$	$[1, 1]$
$supp(ab) = 5$	$[5, 5]$	$[1, 1]$
$supp(ab) = 6$	$[6, 6]$	$[1, 1]$
$supp(bb) = 7$	$[7, 7]$	$[1, 1]$

Therefore, we can conclude that the confidence of $ab \Rightarrow c$ is 1, and hence, derivable.

When the consequent of a rule $X \Rightarrow Y$ consists of more than one item, then its subrules with the same consequent do no longer provide the supports for all necessary subsets of $X \cup Y$. Although we can still derive tight bounds for X using the usual inclusion-exclusion formulas, it becomes a lot more complex to derive the bounds for $X \cup Y$.

EXAMPLE 4. *Consider the rule $abc \Rightarrow de$. From the support and confidence of each of its subrules with the same consequent, we again obtain the support of exactly 2 subsets of $abcde$, i.e., the support of the conditions of the subrules and the support of the sets containing the conditions and the consequent.*

$ab \Rightarrow de$	$ab, abde$
$ac \Rightarrow de$	$ac, acde$
$bc \Rightarrow de$	$bc, bcde$
$a \Rightarrow de$	a, ade
$b \Rightarrow de$	b, bde
$c \Rightarrow de$	c, cde
$\{\} \Rightarrow de$	$\{\}, de$

Hence, apart from the missing supports of the subsets abc and $abcde$, we now also don't have any information on the supports of $d,e,ad,ae,bd,be,cd,ce,abd,abe,acd,ace,bcd$, bce.

Since the consequents of all these rules are the same, we can solve this problem by simply considering the consequent as a single item which occurs in a transaction only if all items in the consequent occur in that transaction. In that way, the problem of multiple items in the consequent is reduced to the case in which only a single item occurs in the consequent, and hence, can be solved as described before.

3.4 Fixed Condition or Consequent We now study the case where the considered subrules have either the same condition or the same consequent as the original rule. The motivation for this approach is that it is likely to be easier for the user to understand redundancy with respect to such subrules than all possible subrules.

To find such non-derivable rules, the first observation is that we can divide the problem into two parts: (1) obtain confidence bounds with fixed consequent subrules, as described in the previous subsection, and with fixed condition subrules (to be described below), and then (2) output the intersection of the possible intervals as the result.

To bound the confidence of $X \Rightarrow Y$ when only those subrules are known that have X as the condition, we need to bound the support of $X \cup Y$, as the support of X is given. To find the bounds, we simply restrict ourselves to those inclusion-exclusion formulas containing only terms that are supersets of X.

3.5 Using Only Some Subrules From an intuitive point of view, it makes sense to measure the value or interestingness of an association rule by comparing to its subrules. As described above, this is exactly what happens when we compute the bounds on the confidence of an association rule using the inclusion-exclusion principle. Unfortunately, for larger sets, the inclusion-exclusion formulas can become quite large and complex, and hence, not so intuitive anymore. Therefore, we also consider the case in which only those subrules with a condition of a minimum size are allowed to be used.

More specifically, for any subset $J \subseteq I$, we obtain a lower or an upper bound on the support of I using one of the formulas in (3.1) or (3.2), but now, we only allow the formulas to be used for those subsets $J \subseteq I$ such that $|I \setminus J| \geq k - 1$, for a user given parameter $k > 0$. We also call this parameter the allowable *depth* of the rules to be used. In Figure 1, the formulas are shown in descending order of depth, starting with depth 5.

In our case we bound not one, but two sets which differ by one in size. We use depth $k - 1$ for the condition of the rule and depth k for all items in the rule.

Dataset	#items	trans. size	#trans.	support threshold	
chess	76	37	3 196	70%	(2238)
connect	130	43	67 557	90%	(60802)
mushroom	120	23	8 124	20%	(1625)
pumsb	7117	74	49 046	85%	(41690)

Table 1: Dataset characteristics

4 Experiments

For an experimental evaluation of the proposed algorithms, we performed several experiments on real datasets also used in [Zak00]. We implemented the proposed algorithms in C++, and for comparison to recent methods we use the original authors' own implementations [LH04, JS02, Zak00, ZP03].

All datasets were obtained from the UCI Machine Learning Repository. The chess and connect datasets are derived from their respective game steps, the mushroom database contains characteristics of various species of mushrooms, and the pumsb dataset contains census data. Table 1 shows some characteristics of the used datasets; for each dataset, we used the lowest support threshold that was mentioned in [Zak00]. The confidence threshold was set to 0% in all experiments.

Figure 2 shows the effect of pruning for the four data sets, as a function of the width of the bound on confidence. Three different variants are shown in each panel (from top to bottom): the number of non-redundant rules when only subrules with identical consequent are used, when only subrules with either identical consequent or identical condition are used, and when all subrules are used. These variants offer different trade-offs between the amount of pruning and how easy it is for the user to understand what was pruned. For a comparison, the number of (minimal) closed rules is also given. (The numbers of minimal closed rules have been obtained with M. Zaki's implementation. They differ from those reported by him in reference [Zak00], since in the latter one he was not exactly mining minimal rules [M. Zaki, personal communication].)

The immediate observation is that pruning has a dramatic effect on the number of rules (note that the Y axis has a logarithmic scale). In particular, a large amount of rules can be derived exactly. Some of the results are also given in numerical form in Table 2. The table reports results for exactly derivable rules with identical consequent subrules, with identical condition or consequent subrules, or with all subrules. The row "1% interval" was obtained by pruning rules for which the lower and upper bounds of confidence are at most 1 percentage point apart. Results with minimal closed rules are included for comparison.

The number of non-derivable association rules is less

	chess	connect	mushroom	pumsb
All rules	8160101	3667831	19245239	1429297
	100%	100%	100%	100%
Identical consequent	1572360	557579	2829208	695871
	19%	15%	15%	49%
Id. condition or consequent	65978	11231	94860	177155
	0.81%	0.31%	0.49%	12%
All subrules	4181	552	7546	16345
	0.051%	0.015%	0.039%	1.1%
All subrules, 1% interval	718	167	5358	543
	0.0088%	0.0046%	0.028%	0.038 %
Minimal closed rules	139431	15496	6815	71813
	1.7%	0.42%	0.035%	5.0%

Table 2: Number of rules after different pruning methods (absolute number and percentage of all rules).

than the number of minimal closed rules already when using only subrules with identical consequent or condition in chess and connect datasets. In pumsb the number of non-derivable association rules is less than the number of minimal closed rules if we use all subrules to compute the upper and lower bound. In mushroom the number of minimal closed rules is slightly less than the number of non-derivable association rules.

Relatively small error bounds, already in the order of fractions of percent, can result in significant further pruning. For example in the mushroom dataset, the number of non-derivable association rules when using all subrules becomes less than the number of minimal closed rules when we allow the difference of upper and lower bound to be one percentage unit. In other datasets the effect of allowing a small interval for the confidence bounds is even more radical.

A comparison to the maximum entropy technique [JS02] and basic association rules [LH04] is given in Figure 3. It shows the number of non-redundant rules with exactly one item in the consequent, since the two other techniques only find redundancies in such rules. A comparison to the maximum entropy approach shows that sometimes it is quite competitive, but it is not a very robust approach for pruning in these cases. The algorithm is approximative and iterative. As a compromise between efficiency and accuracy, we used exactly 5000 iterations in these test; each run then took less than a day except for the chess dataset, for which the execution time was over three days. (The steps visible in some of the maximum entropy graphs are due to a limited accuracy in the output of the implementation, they are not inherent in the method itself.)

The trend seems to be that for very low error bounds, the proposed method is always superior. With a growing error bound, the maximum entropy approach sometimes outperforms non-derivable association rules. The number of basic association rules is considerably greater than the
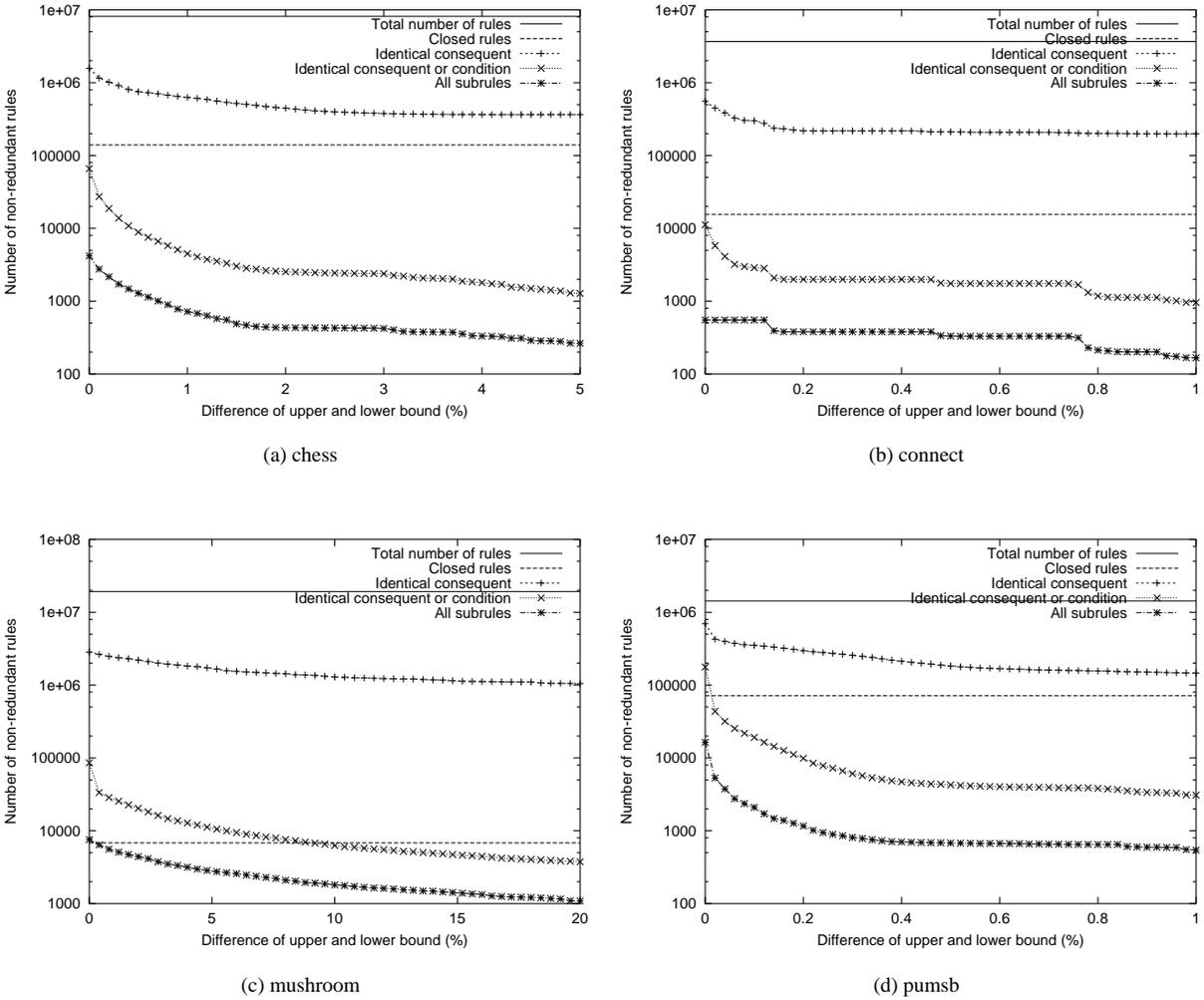
Figure 2: The number of non-derivable and minimal closed association rules.

number of non-derivable rules in all four datasets. As a technique that does not consider error bounds, the basic association rules always outperform the maximum entropy approach in terms of exact inference of rules; sometimes the marginal is quite small, though.

For a further analysis of the proposed method, Figure 4 shows results for different depths of the formulas that were allowed to be used (cf. Section 3.5). This figure only uses association rules with exactly one item in the consequent. The line labeled 'infinite depth' denotes the number of non-derivable rules when all possible formulas are allowed to be used. Additionally, the figure also shows the number of association rules for which the condition is a non-derivable itemset. Since this is a straightforward pruning mechanism based on the notion of non-derivable sets, it shows from where the actual power of the presented confidence derivation tech-

nique starts.

A remarkable result is that most of the derivable rules are already derivable when only the inclusion-exclusion formulas up to depth 3 are allowed to be used. Such a result is particularly nice for the end user, since it means that the reasons for redundancy of a rule are mostly in the most immediate subrules, making the pruning more intuitive and easy to understand.

Finally, Figure 5 shows the number of rules as a function of the support thresholds much lower than those presented in [Zak00]; again with a singular consequent. In these figures, an association rule was considered to be non-redundant if the width of its confidence bound was more than 0.1%. According to the figure, the presented technique scales very well to low support thresholds and achieves roughly similar reductions in the number of association rules across the
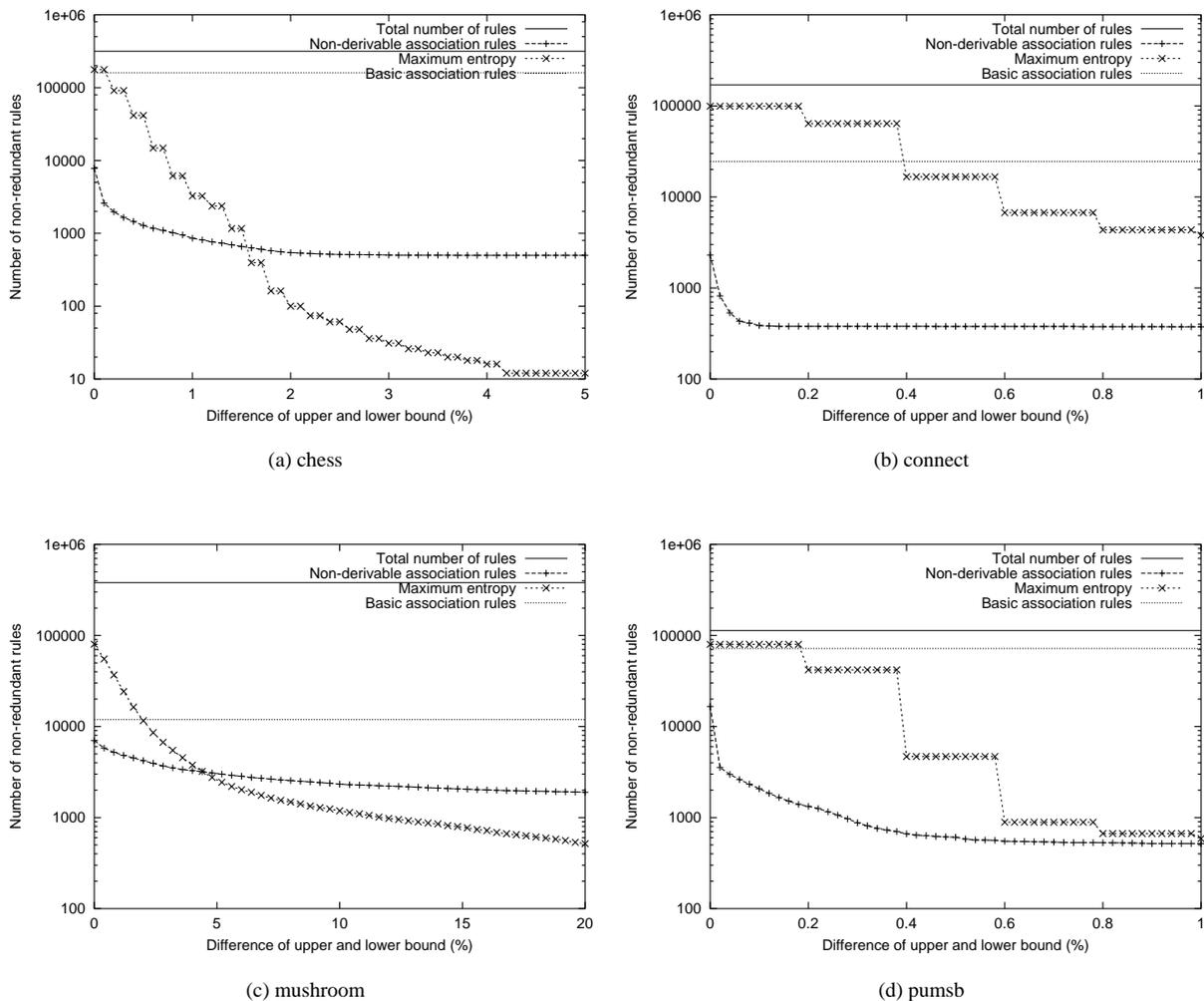
Figure 3: Number of non-derivable and basic association rules and rules produced by maximum entropy method.

ranges tested.

5 Conclusions

We presented a solid foundation for computing upper and lower bounds of the confidence of an association rule, given its subrules. When the upper and lower bounds are equal or almost equal, we call the association rule derivable and consider it to be redundant with respect to its subrules. The presented technique is based on the inclusion–exclusion principle, recently successfully used for bounding the support of sets of items [CG02]. The method is simple, it gives absolute bounds, and it does not assume any specific inference system. The bounds and derivability follow from the definitions of support and confidence: when a rule is pruned as exactly derivable, then there exists only one value for the confidence that is consistent with all the subrules.

Experimental results with real data sets demonstrated very high pruning power. In our experiments, up to 99–99.99% of rules were exactly derivable, and always over 99.96% derivable within ±0.5% points. The amount of pruning depends a lot on data set characteristics as well as on the support threshold: the lower the threshold, the more redundant is the rule set. In absolute terms, the figures indicate great practical significance.

In comparison to related techniques, it is surprising how efficient the proposed simple method is. The related techniques almost invariably make strong assumptions, in the form of fixing an inference system or an estimation method. In the face of the experimental results, our simple and consistent bounding can give much higher pruning factors without any such assumptions.

We gave three different variants of the method, using
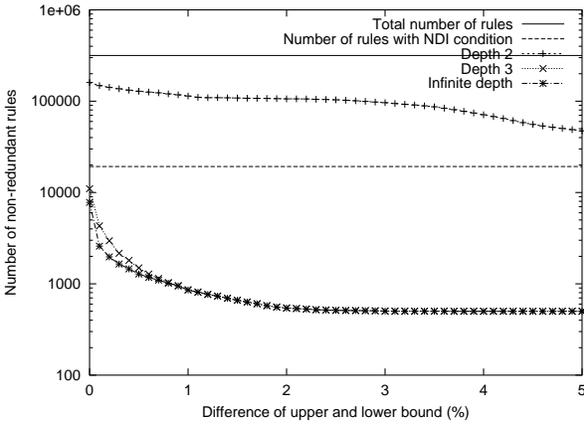
(a) chess
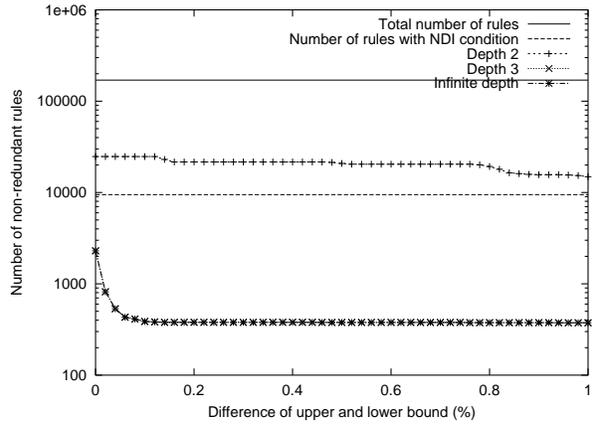
(b) connect
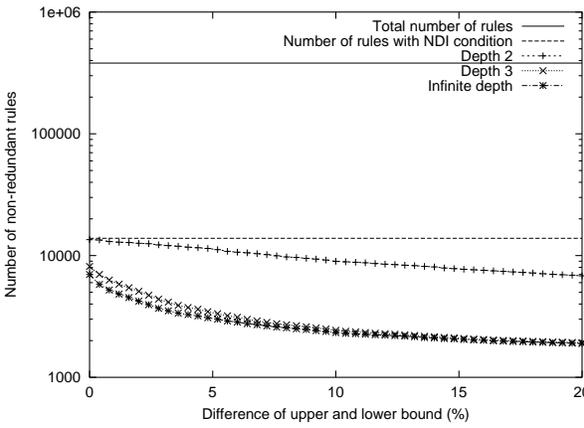
(c) mushroom
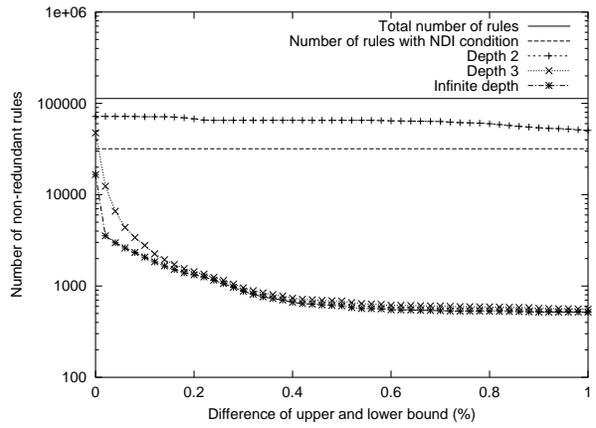
(d) pumsb

Figure 4: The number of non-derivable association rules with a singular consequent.

different sets of subrules to obtain the confidence constraints. They have different trade-offs between the amount of pruning and understandability of pruning. An evaluation of different pruning mechanisms from the end user point of view is a topic for further work.

An important and valid critique on the proposed techniques is that in practice we do not actually have all subrules of an association rule as some of them might not be confident. Indeed, in our experiments, we never used the confidence threshold for pruning, i.e. it was set to 0. Nevertheless, also for higher minimum confidence thresholds, it is always easy to simply compute the actual confidence of all necessary subrules given the frequent itemsets. Furthermore, our experiments show that the numbers of frequent non-derivable association rules are extremely small without using a confidence threshold. Note that in practice, it is not always clear

which confidence threshold should be used and rules with small confidence can sometimes even be extremely interesting.

Nevertheless, in future work, we will explore a sequential pruning mechanism in which only subrules are used that are confident and that where not already pruned earlier.

Acknowledgements

References

[AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami.
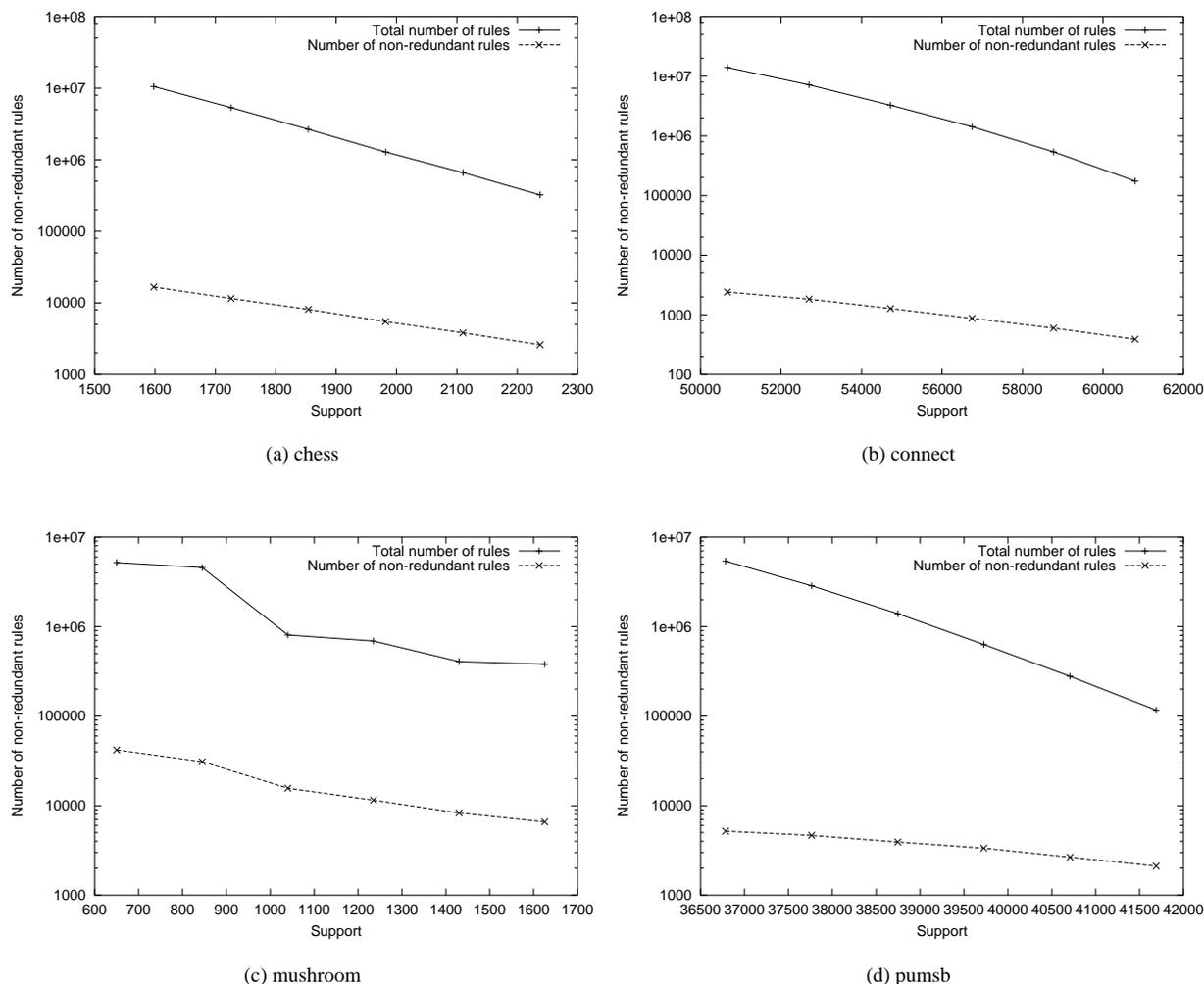
Figure 5: The number of non-derivable association rules for different support thresholds.

Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, 5(6):914 – 925, December 1993. Special Issue on Learning and Discovery in Knowledge-Based Databases.

[BBR00] J-F. Boulicaut, A. Bykowski, and C. Rigotti. Approximation of frequency queries by means of free-sets. In *The Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'00)*, pages 75–85, Lyon, France, 2000. Springer.

[BPT⁺00] Yves Bastide, Nicolas Pasquier, Rafik Taouil, Gerd Stumme, and Lotfi Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. In *Computational Logic – CL 2000: First International Conference*, pages 972 – 986, London, UK, 2000.

[CG02] T. Calders and B. Goethals. Mining all non-derivable frequent itemsets. In T. Elomaa, H. Mannila, and H. Toivonen, editors, *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, volume

2431 of *Lecture Notes in Computer Science*, pages 74–85. Springer, 2002.

[GS00] J. Galambos and I. Simonelli. *Bonferroni-type Inequalities with Applications*. Springer, 2000.

[GVdB00] B. Goethals and J. Van den Bussche. On supporting interactive association rule mining. In Y. Kambayashi, M.K. Mohania, and A.M. Tjoa, editors, *Proceedings of the Second International Conference on Data Warehousing and Knowledge Discovery*, volume 1874 of *Lecture Notes in Computer Science*, pages 307–316. Springer, 2000.

[JS02] S. Jaroszewicz and D. A. Simovici. Pruning redundant association rules using maximum entropy principle. In *Advances in Knowledge Discovery and Data Mining, 6th Pacific-Asia Conference, PAKDD'02*, pages 135–147, Taipei, Taiwan, May 2002.

[KMR⁺94] Mika Klemettinen, Heikki Mannila, Pirjo Ronkainen, Hannu Toivonen, and A. Inkeri Verkamo. Finding interesting rules from large sets of discovered association rules. In

Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94), pages 401 – 407, Gaithersburg, MD, USA, November 1994. ACM.

[LH04] Guichong Li and Howard J. Hamilton. Basic association rules. In *Fourth SIAM International Conference on Data Mining*, Florida, USA, 2004.

[MPS99] Heikki Mannila, Dmitry Pavlov, and Padhraic Smyth. Prediction with local patterns using cross-entropy. In *Proceedings of the ACM SIGKDD*, pages 357–361. ACM Press, 1999.

[NLHP98] R.T. Ng, L.V.S. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimizations of constrained association rules. In L.M. Haas and A. Tiwary, editors, *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, volume 27(2) of *SIGMOD Record*, pages 13–24. ACM Press, 1998.

[PBTL99] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th International Conference on Database Theory*, volume 1540 of *Lecture Notes in Computer Science*, pages 398–416. Springer, 1999.

[Zak00] Mohammed J. Zaki. Generating non-redundant association rules. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 34 – 43, Boston, MA, USA, 2000.

[ZP03] Mohammed Zaki and Benjarath Phoophakdee. MIRAGE: A framework for mining, exploring and visualizing minimal association rules. Technical Report RPI CS Dept Technical Report 03-04, Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York, July 2003.