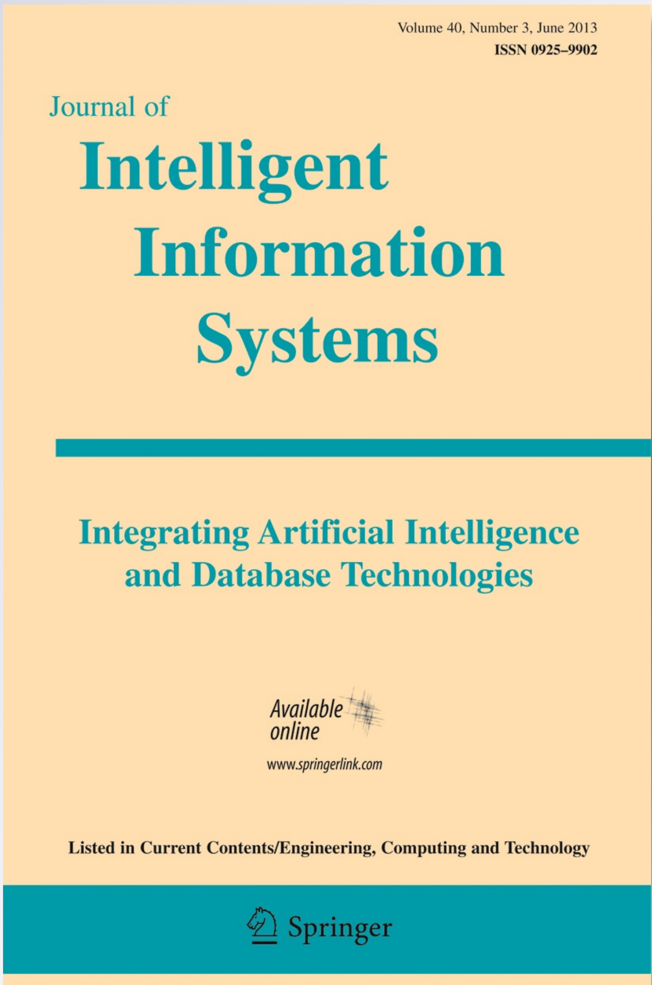
BruteSuppression: a size reduction method for Apriori rule sets

Jon Hills, Anthony Bagnall, Beatriz de la Iglesia & Graeme Richards

Journal of Intelligent Information Systems
Integrating Artificial Intelligence and Database Technologies

ISSN 0925-9902
Volume 40
Number 3

J Intell Inf Syst (2013) 40:431-454
DOI 10.1007/s10844-012-0232-5



Springer

J Intell Inf Syst (2013) 40:431–454
DOI 10.1007/s10844-012-0232-5

BruteSuppression: a size reduction method for Apriori rule sets

Jon Hills · Anthony Bagnall · Beatriz de la Iglesia · Graeme Richards

Received: 19 January 2012 / Revised: 5 December 2012 / Accepted: 6 December 2012 /
Published online: 10 January 2013

Abstract Association rule mining can provide genuine insight into the data being analysed; however, rule sets can be extremely large, and therefore difficult and time-consuming for the user to interpret. We propose reducing the size of *Apriori* rule sets by removing overlapping rules, and compare this approach with two standard methods for reducing rule set size: increasing the minimum confidence parameter, and increasing the minimum antecedent support parameter. We evaluate the rule sets in terms of confidence and coverage, as well as two rule interestingness measures that favour rules with antecedent conditions that are poor individual predictors of the target class, as we assume that these represent potentially interesting rules. We also examine the distribution of the rules graphically, to assess whether particular classes of rules are eliminated. We show that removing overlapping rules substantially reduces rule set size in most cases, and alters the character of a rule set less than if the standard parameters are used to constrain the rule set to the same size. Based on our results, we aim to extend the Apriori algorithm to incorporate the suppression of overlapping rules.

Keywords Apriori · Data mining · Interestingness · Partial classification · Rules

1 Introduction

Association rule mining algorithms generate rule sets from datasets. A rule set may contain hundreds, thousands, or even tens of thousands of rules. In general, many of the discovered rules will be trivial or well-known (for example, associating older age with poorer health for a medical data-mining project), and there may be significant

This work was supported by the UEA Annual Alumni Fund.

J. Hills (✉) · A. Bagnall · B. de la Iglesia · G. Richards
School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, UK
e-mail: j.hills@uea.ac.uk

overlap between rules. Hence, a domain expert must be employed to examine the rule set and identify the interesting rules. This can greatly increase the expense of the data-mining process. One of the advantages of association rule mining is that the models are comprehensible to non-expert users. However, the size of the rule set can decrease the comprehensibility of the model. It is important, therefore, to minimise the size of association rule sets.

The standard approach to reducing association rule set size (used in, for example, Apriori (Agrawal and Srikant 1994)) involves eliminating rules with parameter values (such as support and confidence) that are below certain thresholds. This form of rule reduction has two major drawbacks. Firstly, deleting rules purely because, for example, they do not cover enough instances can mean that interesting rules describing niches in the space of possible cases are removed. Rules that cover few cases but with high accuracy are known as *exception rules* (Liu et al. 1999; Hussain et al. 2000), and are potentially of great interest, as they are more difficult to find by manual analysis than *strong rules* (highly accurate rules covering many cases) or *general rules* (less accurate rules covering many cases). The second disadvantage is that the structure of the rule set is highly dependent on the parameter thresholds, and the user has no *a priori* guidance on which values to choose. Recent research has focused on finding other methods to reduce rule set size, either by adjusting the rule induction algorithm (Shaharanee and Hadzic 2011; Xu et al. 2011), incorporating new interestingness measures (Sarma and Mahanta 2012), or by pruning the rule set (Liu et al. 2011).

We propose an algorithm for reducing the size of rule sets that is based on deleting rules that overlap with other rules. The basic intuition behind this approach is that if two rules are very similar in the cases they cover, then it is safe to delete one of them (Reynolds and de la Iglesia 2006) discuss the case in which two semantically different rules cover the same set of records; whether such rules are interesting is beyond the scope of this paper). In contrast, if a rule uniquely covers a set of cases, then it is worth retaining even if it has lower support and confidence than other rules in the rule set.

We assess the overlap between rules using a measure based on *suppression* (Gebhardt 1991). We construct the reduced rule set through a specifically-tailored comparative enumeration of the rules in the original set. We evaluate our *Brute-Suppression* algorithm on five data sets, and show that the rule sets reduced by BruteSupression maintain the important characteristics of the original rules more effectively than the sets reduced by parameter setting.

We make three main contributions:

1. We define the BruteSuppression algorithm for reducing rule set size by removing overlapping rules.
2. We propose two novel interestingness measures, *swing* and *swing surprisingness*, that incorporate information about the attributes that make up the rule.
3. We present experimental results showing that BruteSuppression causes less disruption to the rule set structure than threshold setting.

The structure of the paper is as follows: Section 2 contains background and related work. In Section 3 we describe the BruteSuppression algorithm, and in Section 4, we document the results of our experiments. We conclude (Section 5) that removing overlapping rules produces smaller, more comprehensible rule sets without

Table 1 An example dataset

Type	Colour	Age	Claim
Estate	Red	5	No
Saloon	Blue	8	No
Sports	Green	1	Yes
Compact	Blue	5	Yes

structurally altering the original rule set. We aim in future work (Section 5.1) to embed BruteSuppression within the Apriori algorithm. The datasets and additional results for our experiments can be found on the companion website to this paper (Paper Authors 2012).

2 Background

2.1 Association rules

Classification is a low-level data-mining task aimed at labelling each record in a dataset as belonging to one of a number of pre-defined classes. The main goal of *partial classification* is to discover rules that reveal characteristics of some pre-defined class(es); such rules need not cover all classes or all records in the dataset (Ali et al. 1997; de la Iglesia et al. 2006). *Association rules* can be used for partial classification. Association rules take the form:

$$Antecedent \Rightarrow Consequent$$

and indicate the degree of co-occurrence between the antecedent conditions and the consequent. In the case of partial classification, the consequent is the class label. Association rules used in this way can provide insight into the characteristics that co-occur within a given class.

We begin with a number of definitions. A dataset, $D = \{x_1, ..., x_n\}$, can be viewed as a set of n instances, where each instance $x_i =< A_1, ..., A_m >$ is an ordered set of m values. Each A_i represents the value of the ith *attribute* for that instance. Each attribute has a range of acceptable values, which may be categorical (e.g. $\{RED, BLUE, GREEN\}$), or continuous.[1] (e.g. the integers 1–100) Table 1 shows a dataset consisting of four instances and four attributes.

An *attribute test* (AT; see, e.g., Richards and Rayward-Smith (2005)) is represented as an ordered triple, $< ATT, OP, VAL >$, where ATT is one of the attributes of the records in the data set, OP is one member of the set $\{<, \leq, >, \geq, =\}$,[2] and VAL is a permissible value for the attribute. The notation $|AT|$ is used to indicate the number of records in the dataset that satisfy that AT. For any AT, the support for that AT ($Sup(AT)$) is calculated as follows:

$$Sup(AT) = |AT|$$

[1] The implementation of Apriori used for our experiments is restricted to categorical attributes.

[2] Where ATT is categorical, '=' is the only applicable OP.

For example, for the dataset shown in Table 1, two records satisfy the AT $\{Colour = Blue\}$, so $Sup(\{Colour = Blue\}) = 2$. The support for a conjunction of ATs is the number of records that satisfy all of the conjuncts:

$$Sup(AT_1 \wedge AT_2) = |AT_1 \wedge AT_2|$$

For the dataset in Table 1, a single record satisfies the conjunction of ATs $\{Colour = Blue\} \wedge \{Type = Saloon\}$.

Association rules take the form $Antecedent \Rightarrow Consequent$ ($A \Rightarrow C$), where both the antecedent and consequent of the rule are some conjunction (or disjunction) of ATs. An association rule for the dataset shown in Table 1 might be:

$$\{Type = Sports\} \Rightarrow \{Claim = Yes\}$$

The association rules studied here have a fixed consequent consisting of a single attribute test with a pre-defined value; hence, they may be used for partial classification.

For any association rule $R = A \Rightarrow C$ (where A and C are conjunctions of ATs representing the antecedent and consequent of the rule respectively), the support for the rule ($Sup(R)$) is calculated as follows:

$$Sup(R) = |A \wedge C|$$

which is the number of records in the dataset that satisfy both the antecedent and the consequent of the rule. The confidence of R (denoted $Conf(R)$) is:

$$Conf(R) = \frac{Sup(A \Rightarrow C)}{Sup(A)}$$

The confidence of a rule is the support for the rule divided by the support for the antecedent.

Consider the dataset shown in Table 1. Take R_1 to be the rule $\{Colour = Blue\} \Rightarrow \{Claim = Yes\}$. $Sup(R_1) = 1$, as only one record satisfies both the antecedent and the consequent of the rule. $Conf(R_1) = 0.5$, as two records satisfy the antecedent of R_1 ($\{Colour = Blue\}$), and $Sup(R_1) = 1$, so $Conf(R_1) = \frac{1}{2} = 0.5$.

Association rules also have a *coverage* value (see, e.g., Major and Mangano (1995)). The coverage of R ($Cov(R)$) is calculated as:

$$Cov(R) = \frac{Sup(R)}{Sup(C)}$$

The coverage of a rule is the support for the rule divided by the support for the consequent, and represents the proportion of records satisfying the consequent that are correctly covered by the rule. Our example rule, R_1, has coverage 0.5, as two records satisfy the consequent ($\{Claim = Yes\}$); $Cov(R_1) = \frac{1}{2} = 0.5$.

2.2 Apriori

We use the *PASW-modeller 14* implementation of Apriori (Agrawal and Srikant 1994) to generate rule sets for our experiments (see Algorithm 1). The Apriori algorithm operates on *itemsets*. An itemset is a combination of *items*, where each item is an AT. Itemsets are arbitrarily ordered (this ordering is used in the generate-Candidates procedure). We denote the kth AT as I_k. This notation is used because

Apriori was originally formulated to solve the *market basket problem* (see Agrawal et al. (1993)), where the antecedent of a rule is a set of boolean values indicating the presence of items in a retail transaction. The algorithm first determines which itemsets are *large* (above the minimum support constraint) and their support (it is common to use a proportion or percentage for the minimum support parameter; in Algorithm 1, *minsup* is assumed to be a count, so proportions, for example, need to be multiplied by the number of records). To determine the large itemsets, Apriori establishes which pairs of items have support above the minimum; these items are retained for the next pass, which finds the sets of three items that have support above the minimum, and so on, until no itemsets have sufficient support (or the maximum number of items has been reached). The maximum support of any itemset containing n items $\{I_1, I_2, ..., I_n\}$ is $Min(Sup(I_1), Sup(I_2), ..., Sup(I_n))$, so this approach is computationally more efficient than assessing every possible itemset. Itemsets are pruned if they have any subset that has not appeared in a previous pass, further reducing the final set of large itemsets, denoted L_{Tot}.

The generateCandidates procedure produces candidate itemsets (C_k) of k items from the set L_{k-1} (described below as Procedure 1). In the *join* stage, any $k-1$ itemsets that differ only in their $k-1$th item are combined to form a k itemset including the $k-1$th item of both itemsets. In the *prune* stage, any itemset generated in the join stage that includes a $k-1$ itemset not included in the set L_{k-1} is removed from the set C_k. The set is returned, the support is calculated, and the set of large k-itemsets, L_k is formed from those candidate itemsets exceeding the minimum support (Agrawal and Srikant 1994).

Algorithm 1 Apriori

```
Input: L_1, the set of large 1-itemsets, D, the set of all instances, minsup, the minimum
  support parameter, minconf, the minimum confidence parameter
Output: RS, the set of all rules above minsup and minconf
  L_1 = {large 1-itemsets} // Generate all large 1-itemsets
  for (k = 2; L_{k-1} ≠ ∅; k++) do
    C_k = generateCandidates(k, L_k − 1) // New candidate itemsets
    for all candidate itemsets c ∈ C_k do
      c.count = 0
    for all instances t ∈ D do
      C_t = C_k ∩ P(t) // Candidate itemsets that are subsets of t
      for all candidate itemsets c ∈ C_t do
        c.count++
    L_k = {c ∈ C_k : c.count ≥ minsup}
  L_Tot = ⋃_k L_k
  RS = ∅; // Generate rule set
  for all large k itemsets l_k ∈ L_Tot where k ≥ 2 do
    H = {consequents of rules derived from l_k with one item in the consequent}
    RS = RS ∪ generateRules(k, l_k, 1, H)
  return RS
```

Once the set L_{Tot} is established, rules are generated by the procedure generateRules (Procedure 2). For each large itemset (l_k), every subset a produces a rule $a \Rightarrow l \sim \{a\}$,[3] which is added to the rule set (RGR) if the confidence of the rule exceeds the $minconf$ parameter. The algorithm shown here will generate all

[3]We use the notation $A \sim B$ to indicate the set $\{x : x \in A \wedge x \notin B\}$.

association rules above the minimum support and confidence thresholds. To generate association rules with a fixed, single AT consequent, we generate only itemsets that contain the consequent, and replace the generateRules procedure with an assessment of the confidence of the rule $R = I \sim \{c\} \Rightarrow c$, where I is the itemset and c is the item representing the fixed consequent.

Procedure 1 generateCandidates

Input: k, L_{k-1}, the set of all large $k-1$-itemsets
Output: C_k, a superset of the set of all large k-itemsets

```
  C_k = ∅
  for all X ∈ L_{k-1}, Y ∈ L_{k-1} do
    if (X ~ {X_{k-1}} = Y ~ {Y_{k-1}}) ∧ (X_{k-1} ≠ Y_{k-1}) then
      c = X ∪ Y
      C_k = C_k ∪ {c}
  for all c ∈ C_k do
    for all k − 1 subsets of c, s do
      if s ∉ L_{k-1} then
        C_k = C_k ~ {c}
  return C_k
```

Procedure 2 generateRules

Input: k, l_k, a k-itemset, m, H, a set of consequents
Output: RGR, the set of all rules satisfying the $minconf$ and $minsupport$ parameters

```
  RGR = ∅
  if k > m then
    m++
    for all h ∈ H do
      conf = support(l_k)/support(l_k ~ h)
      if conf ≥ minconf then
        r =< rule, conf, support(l_k) >, where rule = (l_k ~ {h}) ⇒ h
        RGR = RGR ∪ {r}
      else
        H = H ~ {h}
    H = generateCandidates(H)
    RGR = RGR ∪ generateRules(k, l_k, m, H)
  return RGR
```

The minimum support and minimum confidence parameters are used to reduce the size of the rule set. Rules are eliminated based on counts; this is in contrast to the instance-based BruteSuppression method we advocate. We assess the changes that occur when rule sets are constrained using these parameters, and compare them with the changes caused by our approach.

2.3 Rule quality

We assess rule set quality in terms of the quality of the individual rules in the set. The term typically used in the literature for the quality of a rule is *interestingness*, and we shall adopt this convention. Many measures of rule interestingness have been proposed. These include support, confidence, and coverage (see Section 2.1), *novelty* (Lavrač et al. 1999), *relative risk* (Ali et al. 1997), *chi-square* (Goodman and Kruskal 1954; Bayardo Jr. and Agrawal 1999), *gain* (Fukuda et al. 1996), and *k-measure* (Ohsaki et al. 2004). The majority of interestingness measures are based on counts, such that different rules that happen to have the same counts have the same value. Over 100 of these measures are documented; however, in Bayardo Jr. and Agrawal (1999), the authors demonstrate that many different measures impose the same partial ordering on a rule set. In Hills et al. (2012), the authors show that, for the class of rules that interest us here, a number of popular rule interestingness measures are monotonic with respect to confidence. Hence, for our purposes, there is no benefit in using a large variety of count-based interestingness measures, as we are concerned with their application in terms of rule sets. Of the count-based measures proposed in the literature, we use only confidence and coverage to assess the interestingness of a rule.

3 The BruteSuppression algorithm

In this section, we outline two new interestingness measures, swing and swing surprisingness, and describe our algorithm for reducing rule set size, *BruteSuppression*.

3.1 Swing and swing surprisingness

In contrast to count-based interestingness measures, there is another class of interestingness measure that takes into account the ATs that form the rule. The particular intuition we wish to capture is that good rules are rules that improve on the individual predictive power of the ATs in their antecedent. A rule is more surprising (and therefore interesting) if it is composed of ATs that are poor individual predictors of the target class. To this end, we propose two measures: *swing* (an adaptation of *relative surprisingness* (Hussain et al. 2000) and *confidence gain* (Tamir and Singer 2006)) and *swing surprisingness* (an adaptation of *attribute surprisingness* (Freitas 1999)). These measures are particularly well-suited for data mining, as they reveal cases where combinations of poor predictors have yielded a good rule. This phenomenon can be of interest to domain experts, as it is difficult to identify manually, especially with large datasets and rules consisting of many attributes.

In the following definitions, for any rule R, let $AT_i \Rightarrow C$ be the rule where the antecedent is the ith AT of R, and the consequent (C) is the consequent of R.

We define swing as follows:

$$\text{Swing}(R) = \sum_{i=1}^{n} \frac{Conf(R) \times n}{Conf(AT_i \Rightarrow C)}$$

where rule R has n ATs. Swing focuses on the difference in confidence between the ATs in the antecedent taken singly and the rule taken as a whole.

We define swing surprisingness, SS, as follows:

$$SS(R) = \sum_{i=1}^{n} \frac{n}{Conf(AT_i \Rightarrow C)}$$

where rule R has n ATs. Swing surprisingness is inversely proportional to the mean confidence of the ATs that make up the antecedent of the rule. The measure assigns rules a higher value if they have less predictive ATs, irrespective of the confidence of the rule.

Swing and swing surprisingness are closely related measures; as can be seen, $Swing(R) = SS(R) \times Conf(R)$. Hence, a rule of moderate confidence will have lower swing than a more confident rule composed of equivalently good predictors. This is not the case with swing surprisingness.

3.2 Redundancy of rules

Our aim is to test whether removing redundant rules changes the character of the rule set. Hence, we require a measure to assess the redundancy of a rule. Trivially, if $A \Rightarrow C$ and $A \wedge B \Rightarrow C$ have the same confidence, the second rule is redundant with relation to confidence, as the extra AT adds no predictive power to the rule. Equally (see Balcázar 2009), rule $A \wedge B \Rightarrow C$ is redundant with respect to rule $A \Rightarrow B \wedge C$ and also rule $A \wedge B \Rightarrow C \wedge D$ (assuming equivalent levels of confidence). Apriori allows the user to set minimum support and minimum confidence thresholds; rules that fall below these thresholds do not appear in the rule set. This form of rule set reduction does not take into account any information about the records that the rules cover, only the support and confidence counts for each rule. Hence, rules covering a unique set of records may be eliminated, while rules that cover very similar sets of records may be retained.

Here, we take redundant rules to be those rules that overlap to a large degree, in terms of the records they cover, with rules of greater confidence. That is, if two rules cover the same records (to some specified degree), then the rule with lower confidence is redundant. This is an instance-based form of redundancy, in contrast to the count-based method used to eliminate rules in Apriori. There are a number of measures proposed in the literature to calculate this form of redundancy; see, for example, Gebhardt (1991); Cohen et al. (2001); Reynolds and de la Iglesia (2006). An intuitive measure of the overlap of two rules (R and Q) in terms of the records they cover (D_R and D_Q) is:

$$O(R, Q) = \frac{|D_R \cap D_Q|}{|D_R \cup D_Q|}$$

This represents the size of the intersection of the two sets of records divided by the size of the union of the two sets. We select *suppression* (Gebhardt 1991) as the means to identify redundant rules, because it allows us to incorporate this measure of rule redundancy. Gebhardt (1991) defines a metric for assessing the similarity of rules; we use this metric in our algorithm to delete overlapping rules.

The suppression function calculates if one rule is redundant relative to another rule. Rule R suppresses rule Q if:

$$V(Q) < (1 + \epsilon) \times [S(R, Q)] \times V(R)$$

The function requires some measure of rule interestingness (denoted V), a parameter for determining the intensity of the suppression (denoted as ϵ), and some affinity function (denoted $S(R, Q)$) to measure the similarity of the rules. We use 0.1 for ϵ (this is the most intense suppression recommended in Gebhardt (1991)), and $O(R, Q)$ as our affinity function, as we wish to measure similarity in terms of overlapping coverage of records. We use confidence for V, as it is the standard measure of the quality of a rule. Our suppression function is as follows. Rule R suppresses rule Q if:

$$Conf(Q) < 1.1 \times \frac{|D_R \cap D_Q|}{|D_R \cup D_Q|} \times Conf(R)$$

where $|D_R \cap D_Q|$ is the number of records covered by both rule R and rule Q, and $|D_R \cup D_Q|$ is the number of records covered by either rule R or rule Q.

The algorithm we use to apply the suppression function and indicate redundant rules is given in the next section. Although we test the algorithm on rule sets generated by Apriori, it can be adapted to other association rule mining algorithms such as *Dense Miner* (Bayardo et al. 2000) and *All Rules Algorithm* (Richards and Rayward-Smith 2001).

3.3 BruteSuppression

The BruteSuppression algorithm is shown below as Algorithm 2. The algorithm iterates through a rule set, testing pairs of rules with the suppression function and removing rules deemed to be redundant. Due to the nature of the suppression function, a given rule need only be checked against unsuppressed rules of higher confidence. Hence, the sooner a given rule is suppressed, the fewer total comparisons are required. It is important for efficiency to suppress redundant rules as early as possible. In the worst case, where no rules are suppressed, a rule set of n rules requires $\frac{n(n-1)}{2}$ comparisons (this is lower than the computational complexity of the Apriori algorithm itself). However, since rules are not tested against suppressed rules, in many cases far fewer comparisons are required than in the worst case. In practice, we may achieve a high level of suppression (e.g. on the Adult dataset 50–80 % of the rules are suppressed, see Section 4.3); we have implemented the algorithm to maximise the saving from suppressed rules. This is achieved as follows. The BruteSuppression algorithm iterates through a rule set ordered by confidence, beginning at rule r_2, and comparing each r_i against rule r_{i-1}, then r_{i-2}, and so on, until r_i is suppressed or has been compared to r_1. At this point the algorithm moves to the rule after the current rule. If r_i is suppressed, it will be removed from the rule set; the indexing updates accordingly.

Our implementation checks whether r_i is suppressed, rather than what r_i suppresses, on the following grounds. Empirically, we have observed that rules in Apriori rule sets are most often suppressed by the rules immediately preceding them in the confidence ordering, as overlapping rules tend to have very similar confidence values. It is more common for r_i to be suppressed by, say, r_{i-3}, than for it to be suppressed by r_{i-200}. If the overlapping rules in a rule set are distributed randomly, there is no benefit to any specific ordering. However, for the rule sets we have observed, there is a clear reduction in the number of comparisons if the rules immediately preceding a rule in the confidence ordering are the first rules to which that rule is compared.

Consider the following illustrative example. Assume we have a rule set with 50 % suppression, where rule 1 suppresses rule 2, rule 3 suppresses rule 4, etc. If there are ten rules in the rule set, our approach requires 15 comparisons; testing the rules in descending order of confidence against r_i requires 25 comparisons (with no suppression, 45 comparisons are required). This reduction in the number of comparisons scales to larger rule sets where the overlapping rules cluster together, resulting in a substantial saving. Clearly, the saving occurs only if rule sets exhibit the clustering of overlapping rules that we have observed in our experiments. We have reason to believe that this is the case, because the confidence values of overlapping rules tend to be very similar, as may be expected, meaning that such rules are close to one another in the ordered rule set.

We use confidence in the BruteSuppression algorithm, but any rule interestingness measure can be substituted, depending on the goals of the data-mining project. We chose to use confidence, rather than swing or swing surprisingness, because it is the most commonly-used and easily-understood interestingness measure. We use the two new measures to assess the performance of the algorithm, and recognise that they may have a role to play as learning criteria in future iterations of BruteSuppression (see Section 5). We note, however, that our results (Section 4) show that using confidence as the interestingness measure does not have a negative effect on the swing or swing surprisingness of rule sets. Swing surprisingness in particular is sensitive to pruning by minimum confidence, so it is positive that confidence-based suppression does not have the same effect.

Algorithm 2 BruteSuppression

Input: Rule set $R = < r_1, ..., r_n >$, ordered by descending confidence (ID where confidence is equal), Data set D
Output: R, the set of unsuppressed rules

```
i = 2 // Begin the process from the second rule
ε = 0.1
while i ≤ |R| do
    j = i − 1 // The first rule compared to rule i is the previous unsuppressed rule
    suppress = false
    while j > 0 & !suppress do
        if Confidence(r_j)*(|D_rj ∩ D_ri| / |D_rj ∪ D_ri|) * (1 + ε) ≥ Confidence(r_i) then
            R = R ∼ r_i // Rule i is removed from the rule set
            suppress = true
        j − −
    if !suppress then
        i + + // The index is increased only if the previous rule was not suppressed
return R
```

Our method prunes Apriori rule sets; as such, if the state of the dataset on which the rules are based changes, a new rule set must be generated and then, if necessary, pruned with our algorithm to reduce the number of rules and increase comprehensibility. The rule sets do not update dynamically as the data changes; we recognise that this is an interesting problem in its own right, but it is outside of the scope of our current approach, which is proposed as complementary to Apriori, rather than as part of an incremental learning algorithm.

4 Experimental results

4.1 Methodology

We compare reduced rule sets with the original rule sets, reducing them to approximately the same size using the minimum confidence parameter, the minimum

Table 2 Summary of datasets

Name	Atts/Cont.	N(U)Train/Test	Cons.	N(C)	Base rate
Adult	14/6	30,162/12,435	>50K	7,051/3,846	0.249/0.236
CreditApproval	16/6	411/255	+	172/127	0.419/0.498
HouseVotes	17/0	279/156	republican	110/58	0.394/0.372
Mushroom	23/0	3,700/1,944	e	2,310/1,178	0.624/0.606
Tic-Tac-Toe	10/0	600/358	negative	204/128	0.34/0.358

The column *Atts./Cont* gives the total number of attributes and the number of continuous attributes in a dataset, *N(U)* indicates the total number of records in the training and test sets, *Cons.* the target class, *N(C)* the number of records satisfying the target class, and *Base rate* the proportional incidence of records satisfying the target class in the training and test sets.

antecedent support parameter, or the BruteSuppression algorithm. The effects of the reduction mechanisms are assessed in two ways. Firstly, we estimate the impact of the reduction method on the distribution of the rule set in comparison to the complete rule set. Secondly, we assess the effect on potentially interesting rules caused by the changes in distribution. We take it that a rule set reduction method should cause as little loss of potentially interesting rules as possible, and that a markedly different distribution of values caused by the elimination of whole classes of rules is indicative of this loss.

The datasets we use for our experiments are shown in Table 2. All five data sets are available from the UCI machine learning repository (http://archive.ics.uci.edu/ml/datasets.html), and can also be found at Paper Authors (2012). The dataset referred to as 'HouseVotes' is the Congressional Voting Records set. All other datasets use the names from the UCI repository. For each dataset, we remove any records with missing values[4] and randomly partition the data into training (65 %) and test (35 %) sets. All of the classification problems we study are binary problems.

The parameters used for the experiments are summarised in Table 3. For each row in the table, we generate a number of rule sets from the appropriate dataset using Apriori, with the maximum permitted number of ATs set to each value shown in the column *ATs*. Setting the maximum number of ATs to three produces a rule set containing more general rules, i.e. rules with fewer antecedent conditions. Setting the maximum number of ATs to seven produces more specific rules, i.e. rules with more antecedent conditions. For the datasets we use, rule set size increases when the maximum number of ATs increases. Beyond seven, rule set size does not increase substantially. We do not include the results for the intervening values of the maximum ATs parameter (4–6) in this paper; they are consistent with the results we present, and can be found at Paper Authors (2012). We select the parameter settings for the increased min. confidence and increased min. antecedent support rule sets to produce rule sets similar in size to the suppressed set. This way, a fair comparison can be made between the effects of the BruteSuppression algorithm and the effects of the same degree of reduction from the standard parameters. We do not include parameters for the rule sets generated on the Tic-Tac-Toe dataset, as our algorithm

[4]Records with the value '?' in the HouseVotes dataset are not removed, as they do not represent missing data.

Table 3 Parameters for experiments

Data set	ATs	Original settings	Increased min. conf.	Increased min. sup.
Adult	3	Sup. 2 %, Conf. 0.25	Sup. 2 %, Conf. 0.44	Sup. 5 %, Conf. 0.25
Adult	7	Sup. 2 %, Conf. 0.25	Sup. 2 %, Conf. 0.69	Sup. 8 %, Conf. 0.25
CreditApproval	3	Sup. 10 %, Conf. 0.42	Sup. 10 %, Conf. 0.75	Sup. 30 %, Conf. 0.42
CreditApproval	7	Sup. 10 %, Conf. 0.42	Sup. 10 %, Conf. 0.88	Sup. 30 %, Conf. 0.42
HouseVotes	3	Sup. 30 %, Conf. 0.40	Sup. 30 %, Conf. 0.90	Sup. 37 %, Conf. 0.40
HouseVotes	7	Sup. 30 %, Conf. 0.40	Sup. 30 %, Conf. 0.91	Sup. 37 %, Conf. 0.40
Mushroom	3	Sup. 50 %, Conf. 0.63	Sup. 50 %, Conf. 0.93	Sup. 87 %, Conf. 0.63
Mushroom	7	Sup. 50 %, Conf. 0.63	Sup. 50 %, Conf. 0.93	Sup. 87 %, Conf. 0.63
Tic-Tac-Toe	3	Sup. 5 %, Conf. 0.35	N/A	N/A
Tic-Tac-Toe	7	Sup. 5 %, Conf. 0.35	N/A	N/A

The parameters used in the generation of the rule sets are shown in the columns *Original Settings*, *Increased Min. Conf.*, and *Increased Min. Sup.*

is ineffective on these rule sets. We take it that the BruteSuppression algorithm does not yield large reductions in rule set size for rule sets generated on the Tic-Tac-Toe dataset because these rule sets do not contain very many overlapping rules. Using another method to reduce Tic-Tac-Toe rule sets may change the character of the rule set, as is the case with rule sets generated on other datasets.

The minimum confidence settings for the original rule sets are just above the base rate of the consequent in the training set. We use the base incidence rate as the minimum confidence on the assumption that rules with confidence lower than this are uninteresting. The minimum support settings are chosen on a pragmatic basis to prevent an explosion in the number of rules. We feel that this has not compromised the results, however, as the settings correspond to those a user might select, giving us a more realistic idea of the effectiveness of our algorithm. In addition, we have tested the algorithm on a single unconstrained rule set with similar results to the constrained rule sets. Ultimately, we intend for suppression to be used in conjunction with the standard parameters, to allow those parameters to be set at lower levels than they would be otherwise.

We generate rule sets with three ATs and seven ATs to collect data on rule sets containing only short rules (the three AT sets), and rule sets that also contain longer rules (the seven AT sets). All of the rule sets are assessed on previously unencountered test sets.

We examine how suppression affects the distribution of rules on coverage/confidence, confidence/swing, and confidence/swing surprisingness graphs. In particular, we are interested in whether certain classes of rules are eliminated by suppression. For the coverage/confidence graph, we divided rules into four classes: strong (high coverage and high confidence), general (high coverage, low confidence), exception (low coverage, high confidence, see Liu et al. (1999); Hussain et al. (2000)), and weak (low coverage, low confidence). We do not discuss the effect of suppression on weak rules, as these are unlikely to be of interest. We are also interested in the distribution of rules on confidence/swing and confidence/swing surprisingness graphs, as rules of high swing/swing surprisingness relative to their confidence are potentially of interest.

The second area we investigate is the effect of the reduction methods on the distribution of rules in a rule set using the chi-squared statistic. If two sets of

Table 4 Ranking of rule reduction methods by resemblance to original rule set

Rule set	Avg.rank	Rank 1	Rank 2	Rank 3
Suppressed	1.22	25	7	0
Increased minimum confidence	2.38	3	14	15
Increased minimum antecedent support	2.41	4	11	17

The columns indicate the average rank of the reduced rule sets, and the number of instances in which the reduced rule set was ranked first, second, or third.

values are drawn from the same distribution, the chi-squared statistic obtained by comparing them is likely to be small. Large chi-squared values are indicative of the rule sets having different distributions of the qualities in question. We assess the significance of changes at a significance level of 0.01, as well as ranking the different methods by their chi-squared values (see Table 4) for rule confidence, coverage, swing, and swing surprisingness. We discuss the results of our experiments on the Adult dataset in detail (Section 4.3) and summarise the results for the other datasets (Section 4.2, 4.4, see also Paper Authors 2012).

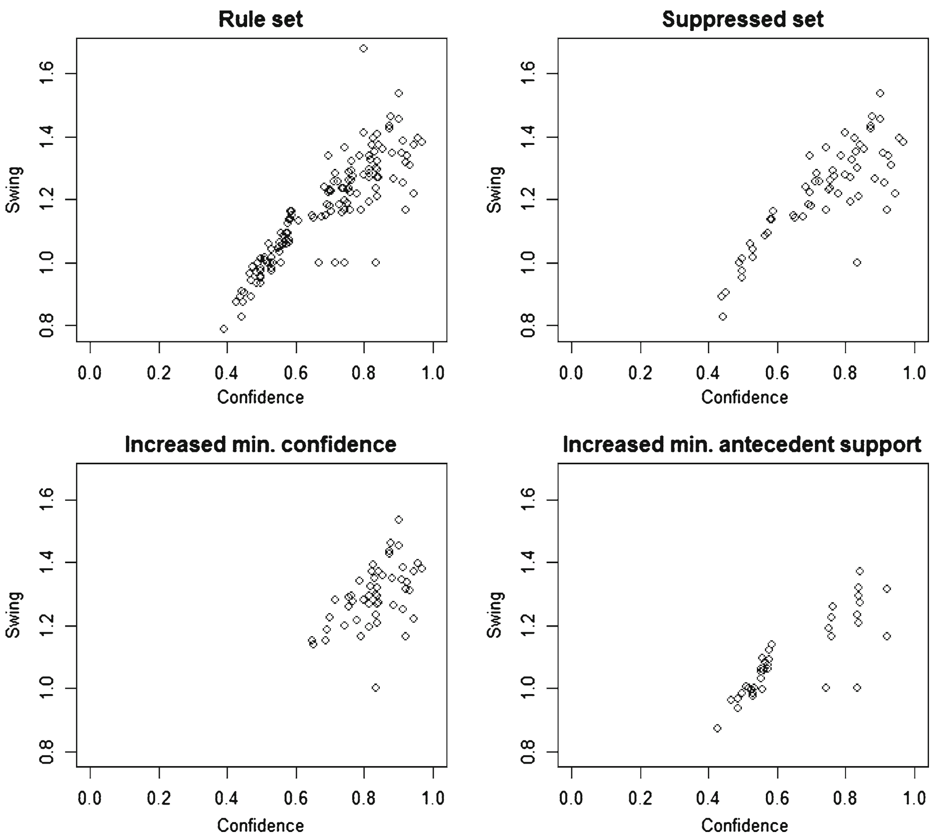


Fig. 1 Confidence/swing rule distribution, CreditApproval 3 ATs

4.2 Summary of results

A summary of the results of our chi-squared distribution tests is shown in Table 4. The hypothesis we test is that, for each reduced rule set, there is no difference in the distribution of confidence, coverage, swing, or swing surprisingness between that rule set and the original, larger rule set. For each reduced rule set, we generate a chi-squared statistic for each of the four properties by comparing the reduced rule set to the original set. The lower the chi-squared statistic (for a given original rule set), the more closely the distribution of rules in the reduced set matches that of the rules in the original set for that property. For each original set, we ranked the three reduced rule sets by their chi-squared statistic, assigning rank one to the reduced rule set with the lowest chi-squared statistic, and hence the greatest resemblance to the original rule set. Table 4 shows the number of instances in which each type of reduced rule set is ranked first, second, and third, and the average rank.

As can be seen in Table 4, the suppressed rule sets had consistently lower chi-squared values when compared to the original rule set than rule sets reduced by the other two methods. For 25 of the 32 original rule sets, the suppressed set bears

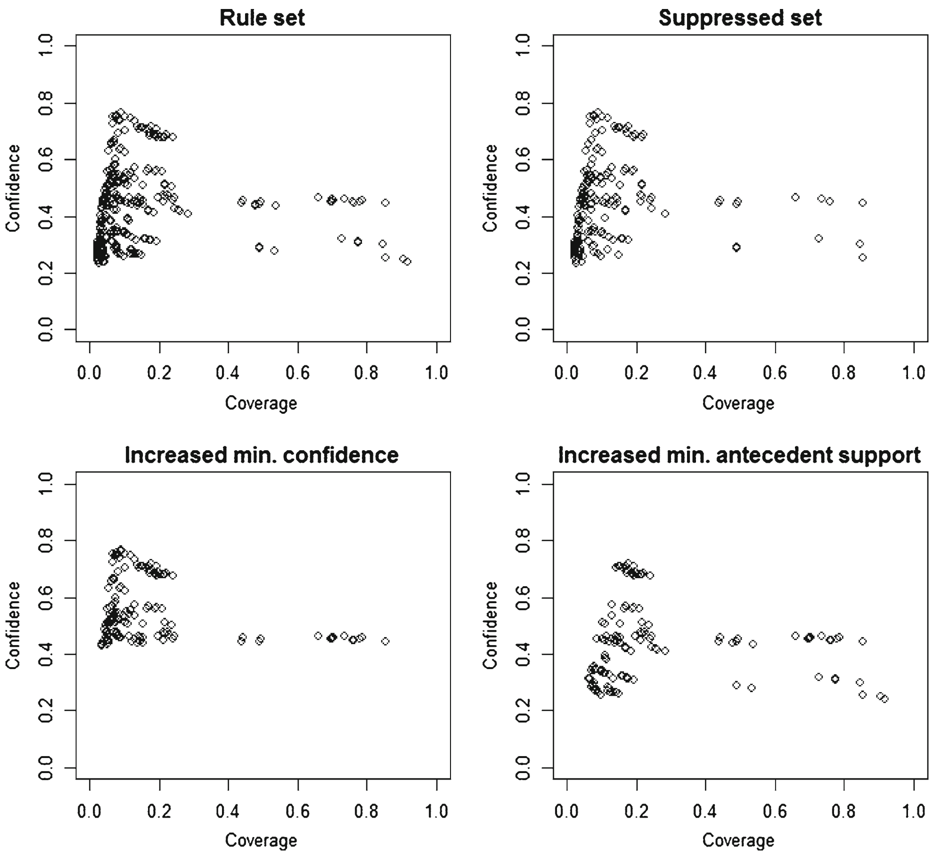


Fig. 2 Coverage/confidence rule distribution, Adult 3 ATs

the closest resemblance to the original set, and in no cases are both of the other methods closer to the original rule set than the suppressed set is. We also examine the distribution of rules visually, an example of which is shown in Fig. 1. The suppressed rule set is the most similar to the original rule set, and it is clear that large classes of potentially interesting rules are eliminated by the other two reduction methods.

We turn now to a more detailed examination of our experimental results.

4.3 Results: Adult dataset

4.3.1 Coverage/confidence distribution

For rule sets generated on the Adult dataset, the distribution of rules on a coverage/confidence graph is consistent between the original rule set and the suppressed rule set (see Figs. 2 and 3). This is the case both for the rule set generated on Adult with a maximum of three ATs (Fig. 2), and where the maximum number of ATs is seven (Fig. 3). Increasing the minimum confidence parameter to constrain the size of the rule set removes all rules with confidence below 0.4 for the 3 AT Adult rule sets. This may be acceptable depending on the requirements of the mining

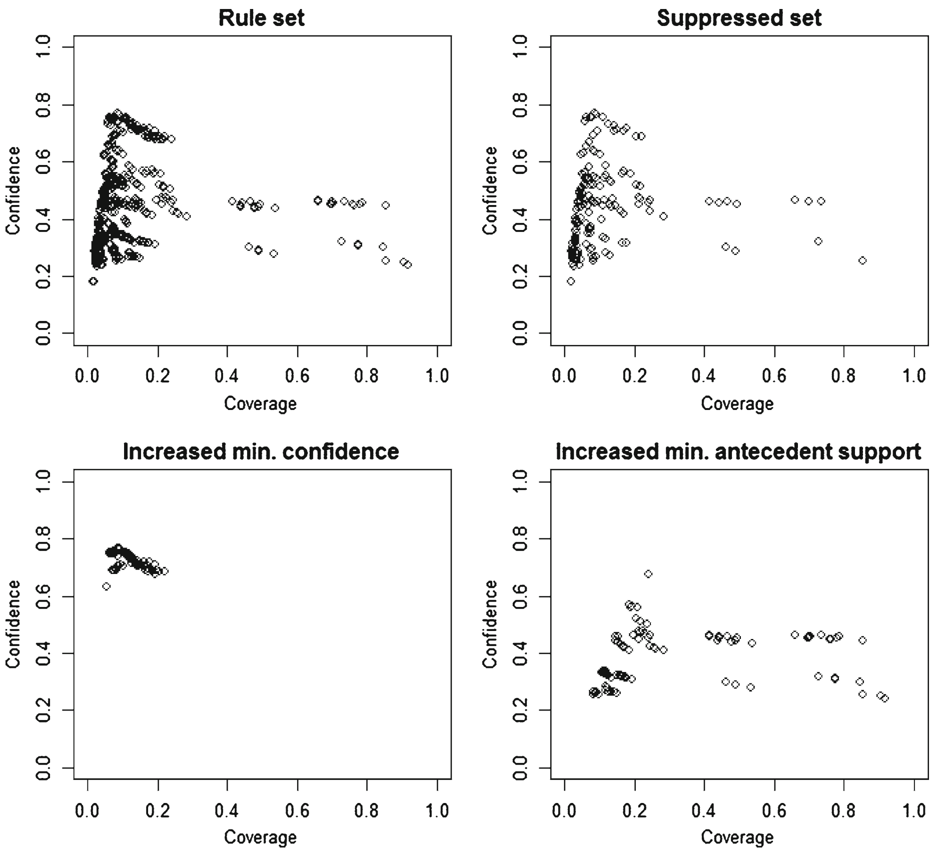


Fig. 3 Coverage/confidence rule distribution, Adult 7 ATs

project. Such rules are not, however, intrinsically uninteresting, and it may be the case that interesting rules (those with confidence between the base rate and 0.4) are removed. This is not the case with the suppressed rule set, as the remaining rules are representative of the original rule set.

The obvious weakness of the 3 AT rule set generated with higher minimum antecedent support is that a large number of exception rules (rules with high confidence but low support) have been eliminated from the set, and these rules can be highly interesting, as they are strong predictors of the target class that are difficult to find by manual analysis.

Other than the differences mentioned above, all three rule sets generated with 3 ATs on the Adult data sets have mostly maintained the original distribution; this is not the case with the rule sets generated on Adult with 7 ATs.

The suppressed rule set generated on Adult with 7ATs has maintained the shape of the original distribution. The rule set constrained with increased minimum confidence has lost every general rule and a large number of exception rules, changing the distribution substantially. The rule set constrained with increased minimum antecedent support has lost all of the higher confidence rules (both exception rules

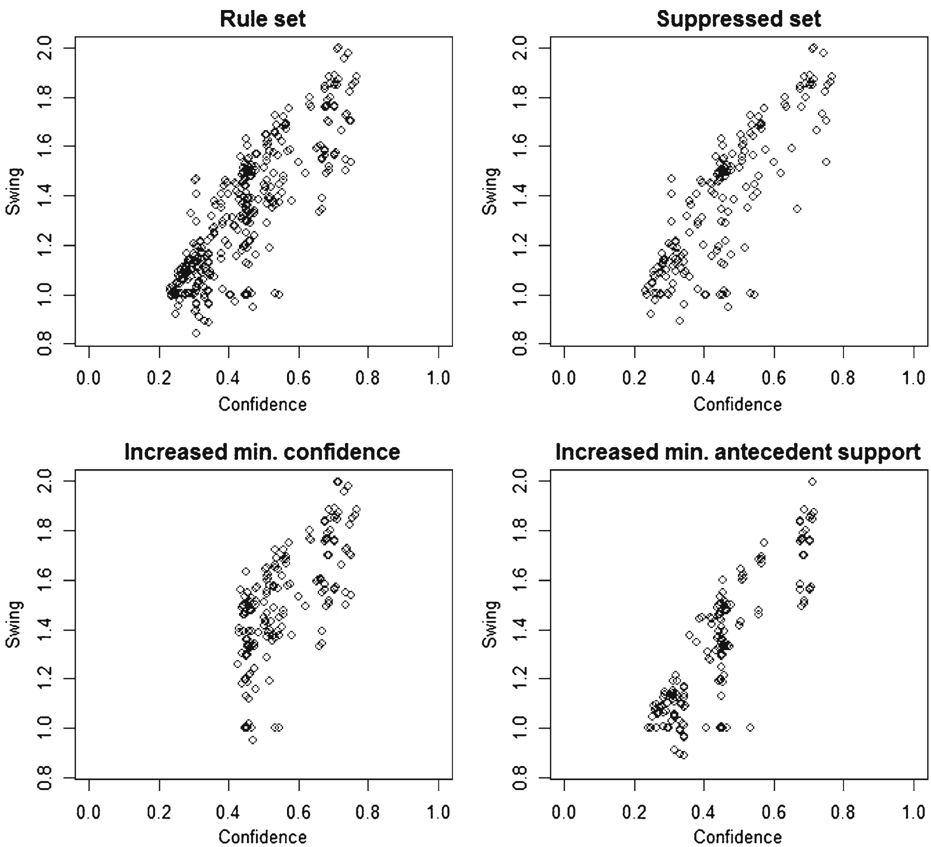


Fig. 4 Confidence/swing rule distribution, Adult 3 ATs

and strong rules). The coverage/confidence distributions suggest that increasing the minimum confidence or antecedent support parameters to constrain rule set size negatively affects the rule set, relative to a suppressed set of similar size.

4.3.2 *Confidence/swing distribution*

The suppressed rule sets generated on Adult with 3 ATs and 7 ATs both maintain the shape of the confidence/swing distribution to a high degree (see Figs. 4 and 5). This suggests that swing is unlikely to be negatively affected by suppression. For the 3 AT rule sets (Fig. 4), the set constrained by increased minimum confidence has all rules of confidence lower than 0.4 removed. This has removed a number of rules with high swing relative to their confidence. The rule set constrained with increased minimum antecedent support has a number of missing rules with high swing relative to their confidence.

For the rule sets generated with 7 ATs (Fig. 5), neither of the sets constrained with the Apriori parameters have maintained the shape of the confidence/swing distribution. Many rules with high swing relative to their confidence have been

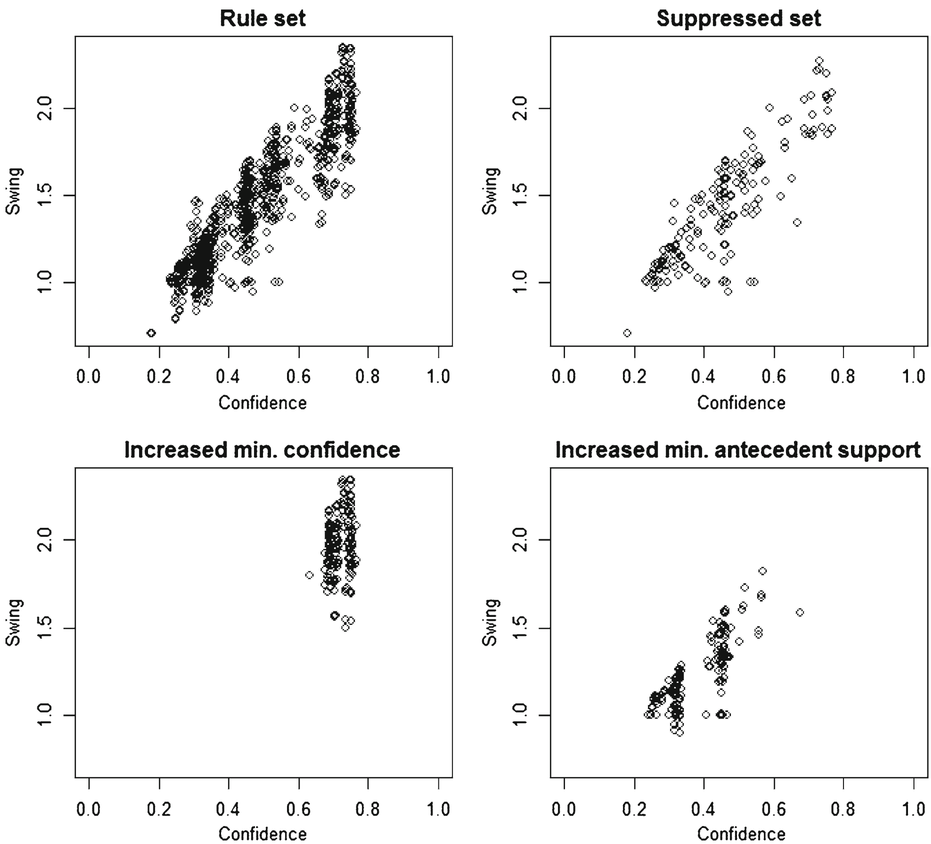


Fig. 5 Confidence/swing rule distribution, Adult 7 ATs

eliminated, suggesting that the rule sets are compromised in terms of swing relative to the original set and the suppressed set.

4.3.3 Confidence/swing surprisingness

The suppressed rule sets generated on Adult maintain the shape of the confidence/swing surprisingness distribution of the original rule set (Figs. 6 and 7). The rule set constrained with increased minimum confidence (Fig. 6) has many rules of high swing surprisingness missing, relative to the original set. The rule set constrained with increased minimum antecedent support (Fig. 6) resembles the original set more closely, with only a few rules of high swing surprisingness missing from the original rule set.

The differences between the rule sets are more marked for the sets generated on Adult using 7 ATs (Fig. 7). The shape of both the increased minimum confidence and the increased minimum antecedent support rule sets has altered considerably relative to the original rule set. This shows that using these measures to constrain rule set size may adversely affect rule sets.

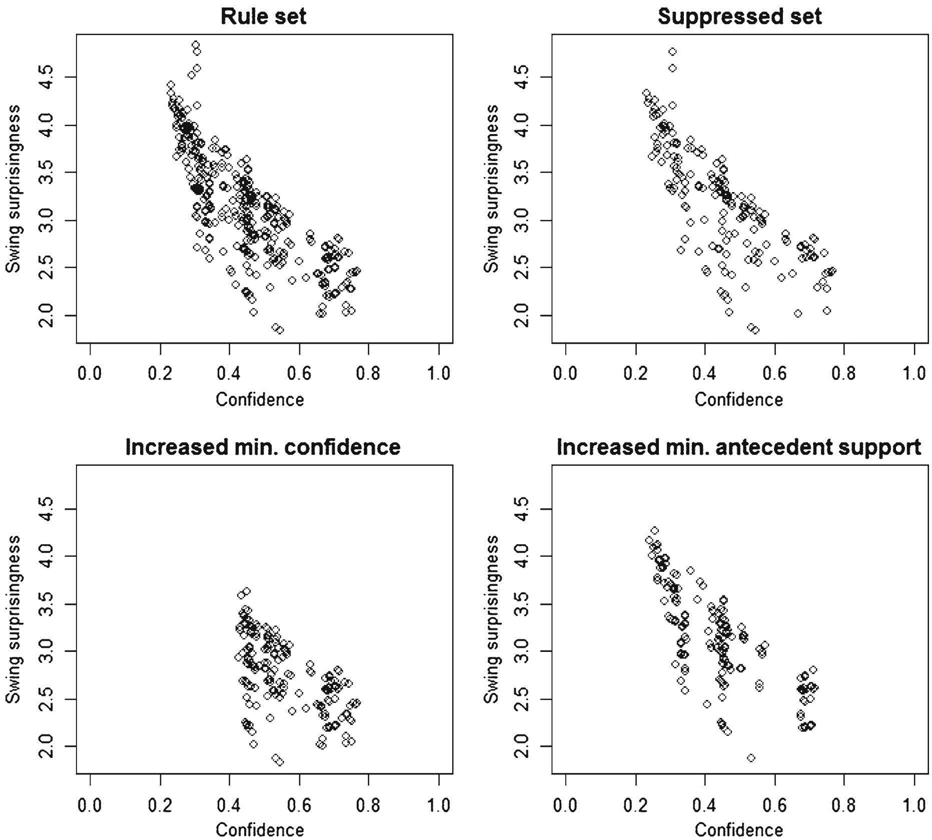


Fig. 6 Confidence/swing surprisingness rule distribution, Adult 3 ATs

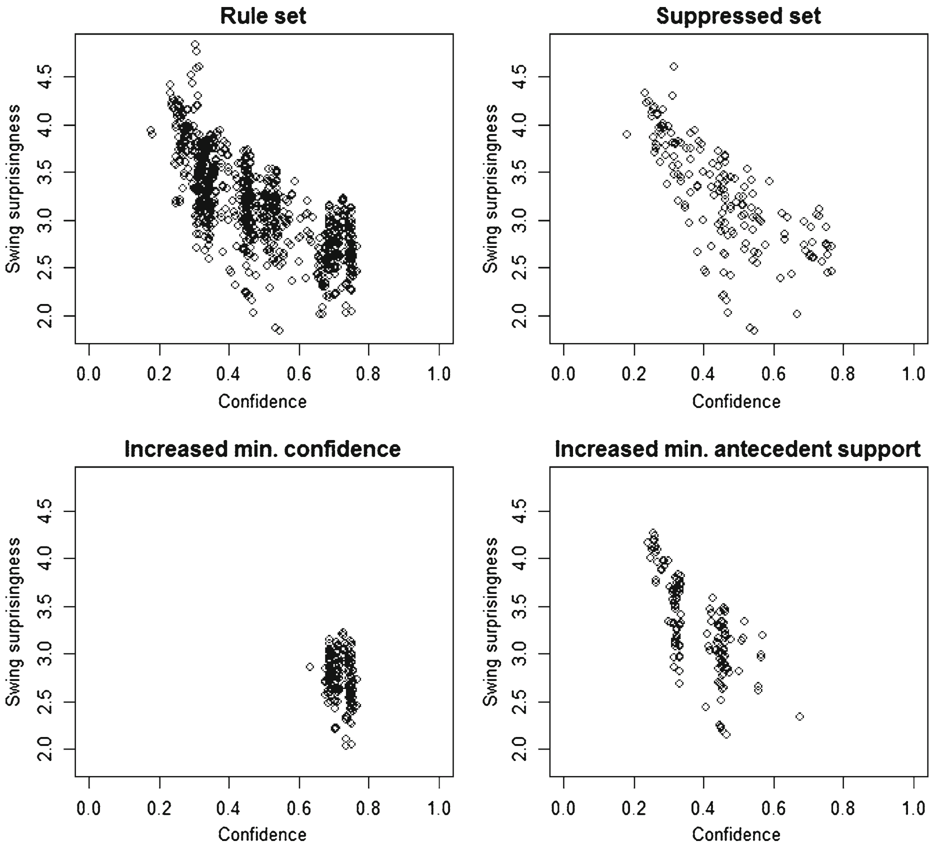


Fig. 7 Confidence/swing surprisingness rule distribution, Adult 7 ATs

4.3.4 Test of distribution

Table 5 shows how the distribution of values for the rule sets have been affected by the three methods of reducing rule set size.

The null hypothesis is that the values in the reduced set are drawn from the same distribution as those in the original set, and the alternative hypothesis is that they are not. We use a chi-squared test, and reject the null hypothesis at a significance level of 0.01. As can be seen from Table 5, reducing rule set size by increasing the minimum confidence causes significant changes in the distribution of confidence, swing, and swing surprisingness of rule sets generated on Adult, as well as significant changes in the distribution of coverage values of rule sets generated with a maximum of 7 ATs. Only the distribution of coverage values for the 3 AT rule set was unchanged from the original set. Reducing rule set size by increasing the minimum antecedent support causes significant changes in the distribution of coverage values for 3 AT and 7 AT rule sets, and in the distribution of confidence, swing, and swing surprisingness values for the 7 AT rule sets. This occurs because the 7 AT rule set contains more rules, so must be constrained with higher parameter settings than the 3 AT sets. The suppressed sets have significantly different distributions

Table 5 Tests of distribution for reduced Adult rule sets compared to original rule set

Rule set	Max. ATs	Measure	χ^2	D.F.	Significant?
Adult suppressed	3	Conf.	5.768	11	No
Adult mod. conf.	3	Conf.	103.233	11	Yes
Adult mod. sup.	3	Conf.	17.797	11	No
Adult suppressed	7	Conf.	43.643	11	Yes
Adult mod. conf.	7	Conf.	360.815	11	Yes
Adult mod. sup.	7	Conf.	105.867	11	Yes
Adult suppressed	3	Cov.	6.534	9	No
Adult mod. conf.	3	Cov.	10.857	9	No
Adult mod. sup.	3	Cov.	73.349	9	Yes
Adult suppressed	7	Cov.	9.133	9	No
Adult mod. conf.	7	Cov.	66.599	9	Yes
Adult mod. sup.	7	Cov.	234.239	9	Yes
Adult suppressed	3	Swing	11.394	11	No
Adult mod. conf.	3	Swing	58.2	11	Yes
Adult mod. sup.	3	Swing.	4	11	No
Adult suppressed	7	Swing	5.098	11	No
Adult mod. conf.	7	Swing	309.183	11	Yes
Adult mod. sup.	7	Swing	63.489	11	Yes
Adult suppressed	3	SS.	1.592	6	No
Adult mod. conf	3	SS	60.711	6	Yes
Adult mod. sup.	3	SS.	4.369	6	No
Adult suppressed	7	SS.	24.653	6	Yes
Adult mod. conf.	7	SS.	150.697	6	Yes
Adult mod. sup.	7	SS.	18.431	6	Yes

The column χ^2 contains the chi-squared statistic for each rule set, the column $D.F.$ shows the degrees of freedom, and the column *Significant*? indicates whether any difference in distribution is significant at 0.01.

only for confidence and swing surprisingness of the 7 AT rule sets. For both of these measures, all three reduction methods make a significant difference to the distribution of values. Our results show that, of the three rule set reduction methods tested, the BruteSuppression algorithm has the least effect on the rule set, for rule sets generated on the Adult dataset.

4.3.5 Size of rule sets

The BruteSuppression algorithm with $\epsilon = 0.1$ reduces rule set size by approximately 49 % for rule sets generated on Adult with 3 ATs, and approximately 81 % for rule sets generated on Adult with 7 ATs. Similar reductions in size were also achieved using the minimum confidence and minimum antecedent support parameters (see Table 6).

4.4 Other datasets

For brevity, we exclude the distribution graphs for the remaining datasets (CreditApproval, HouseVotes, Mushroom, Tic-Tac-Toe). The graphs for three of the datasets (CreditApproval, HouseVotes, and Mushroom) can be found in Paper Authors (2012). For rule sets generated on the Tic-Tac-Toe dataset, suppression is ineffective

Table 6 Size of Adult rule sets

Rule set	Max. ATs	Size before	Size after	Percentage reduction
Adult Suppressed	3	343	176	48.7
Adult mod. conf.	3	343	174	49.3
Adult mod. sup.	3	343	164	52.2
Adult suppressed	7	880	169	80.8
Adult mod. conf.	7	880	175	80.1
Adult mod. sup.	7	880	157	82.2

at reducing rule set size; we conclude from this that suppression does not work on every rule set, as some rule sets will not contain overlapping rules. Given that this is the case, we conclude that, for our instance-based approach at least, the vast majority of the rules are required to cover the Tic-Tac-Toe data. Hence, BruteSuppression eliminates only small numbers of rules and does not compromise the coverage of the rule set.

The distribution graphs show similar characteristics across the Adult, CreditApproval, HouseVotes, and Mushroom datasets. The suppressed rule sets retain the shape of the original rule set; to some extent, this is also the case for rule sets generated with increased minimum antecedent support and 3 ATs. The main difference is that the sets with increased minimum antecedent support are prone to excluding exception rules, and certain rules with high swing/swing surprisingness relative to their confidences. When generated with 7 ATs, the minimum antecedent support constraint must be increased to compensate for the larger size of the rule set. This results in rule sets with very different distributions to the original sets, and many potentially interesting rules missing.

The rule sets generated with increased minimum confidence lose all rules with lower confidence than some threshold. This means that many rules with confidence above the base rate (i.e. rules that are potentially interesting to domain experts) are lost. The distributions are changed dramatically for the 7 AT sets, as these sets require a higher minimum confidence to prevent an explosion in rule set size. These rule sets omit numerous potentially interesting rules.

The major difference between the suppressed sets and the original set appears to be that suppression affects general rules (low confidence and high coverage) the most, so some suppressed rule sets have fewer general rules. This is a weakness of the BruteSuppression method, and could be countered by using coverage, for example, as the rule interestingness measure for the suppression function (assuming that general rules are the main goal of the project). Overall, suppressed rule sets offer a smaller rule set that retains the character of the rules in the original set, in contrast to rule sets generated using increased minimum confidence or antecedent support.

Table 7 lists counts of cases where the reduced set differed significantly in its distribution of values (using a chi-squared test with a significance level of 0.01) from the original set (the results from Adult are included for completeness). Overall, as is evident, the suppressed rule sets display far fewer significant differences than the reduced rule sets generated using the other two methods. This suggests that suppression is the rule reduction method that least affects the distribution of values in Apriori rule sets. Suppression performs most poorly on rule sets generated on the HouseVotes dataset; in this case, the performance is similar to that of the other two

Table 7 Counts of significant differences at 0.01 between reduced rule sets and the original rule set

Data set	Suppressed	Increased min. confidence	Increased min. antecedent support
Adult	2/8	7/8	5/8
CreditApproval	0/8	7/8	5/8
HouseVotes	5/8	4/8	6/8
Mushroom	1/8	8/8	7/8
Total	8/32	26/32	23/32

The total number of possible differences is 32, as there are two rule sets, four values (confidence, coverage, swing, and swing surprisingness), and four datasets ($2 \times 4 \times 4 = 32$).

methods. For the other datasets, suppression has only three significant differences out of a possible 24.

Our results show that, of the three methods for reducing rule set size, suppression has the least effect on the distribution of values for rule sets generated on the Adult, CreditApproval, and Mushroom datasets, and performs no worse than the other methods on rule sets generated on the HouseVotes dataset. This is consistent with the distribution patterns, which show that suppression causes the least change in the character of the rule sets. Where the suppression algorithm is effective, it yields rule sets that share the characteristics of the original rule set. This is not the case for the other two methods used to reduce rule set size. Rule sets sizes are shown in Table 8. Although our methodology is different to that in Liu et al. (2011), we achieve a very similar degree of size reduction for association rule sets derived from the Mushroom dataset, which is common to both papers; BruteSupression gives a 93.5 %–96.9 % reduction, compared to the method in Liu et al. (2011), which yields a 96.3 % reduction.

Table 8 Size of rule sets

Rule set	Max. ATs	Size before	Size after	Percentage reduction
CreditApproval suppressed	3	178	61	65.7 %
CreditApproval mod. conf.	3	178	62	65.2 %
CreditApproval mod. sup.	3	178	52	70.8 %
CreditApproval suppressed	7	388	61	84.3 %
CreditApproval mod. conf.	7	388	57	85.3 %
CreditApproval mod. sup.	7	388	58	85.1 %
HouseVotes suppressed	3	269	63	76.6 %
HouseVotes mod. conf.	3	269	58	78.4 %
HouseVotes mod. sup.	3	269	58	78.4 %
HouseVotes suppressed	7	428	61	85.7 %
HouseVotes mod. conf.	7	428	67	84.3 %
HouseVotes mod. sup.	7	428	58	86.4 %
Mushroom suppressed	3	184	12	93.5 %
Mushroom mod. conf.	3	184	8	95.7 %
Mushroom mod. sup.	3	184	14	92.4 %
Mushroom suppressed	7	519	12	97.7 %
Mushroom mod. conf.	7	519	16	96.9 %
Mushroom mod. sup.	7	519	16	96.9 %

5 Conclusions

We tested the effect of removing overlapping rules from Apriori rule sets (suppression), and compared the rule sets to similarly-sized sets reduced using the minimum confidence and minimum antecedent support parameters. We discovered that the suppressed sets have the strongest resemblance to the original rule sets. Increasing the minimum confidence parameter removed many potentially interesting rules of lower confidence, while increasing the minimum antecedent support tended to eliminate the exception rules, a potentially interesting class of rules.

Removing overlapping rules did not effectively decrease the size of the rule sets generated on the Tic-Tac-Toe dataset. This suggests that it is not a suitable method for reducing the size of every rule set. In contrast, increasing the minimum confidence or minimum antecedent support reduces rule set size for any rule set (within certain bounds), but also has a strong effect on the character of the rule set.

We demonstrated that removing overlapping rules can reduce rule set size without altering the character of a rule set, or removing classes of potentially interesting rules. BruteSuppression could be used in combination with suitable Apriori parameter settings to ensure that rule sets are small enough to be comprehensible, without the loss of potentially interesting rules that occurs if the Apriori parameter settings are used alone to reduce rule set size.

5.1 Future work

The primary application of this work in the future will be to produce a modified version of Apriori that incorporates suppression to avoid generating overlapping rules. Using the current BruteSuppression algorithm involves post-processing the rule set. By incorporating suppression into the association rule mining process, smaller rule sets can be generated that are representative of the rule sets that would otherwise be generated. By definition, suppression only eliminates rules with a high degree of overlap with other rules, and we have shown empirically that the rule sets are altered more by the standard Apriori parameters than they are by removing the overlapping rules.

Having established the efficacy of BruteSuppression based on the well-known confidence measure, we intend to test different interestingness measures in place of confidence in the suppression function, particularly our new measures, swing and swing surprisingness. It might be the case that better results can be obtained with these functions, or that different qualities of the rule set can be emphasised by using a function based on, for example, coverage, rather than confidence. Additionally, it may be possible to implement a suppression ensemble, with different suppression functions contributing to the algorithm.

A final area to examine is the effect of using values of ϵ greater than 0.1, to investigate how this affects rule set size and composition.

References

Agrawal, R., Imieliński, T., Swami, A. (1993). Mining association rules between sets of items in large databases. In *ACM SIGMOD record* (Vol. 22, pp. 207–216). ACM.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB* (Vol. 1215, pp. 487–499).

Ali, K., Manganaris, S., Srikant, R. (1997). Partial classification using association rules. In *Proceedings of the third international conference on knowledge discovery and data mining* (pp. 115–118).

Balcázar, J. (2009). Confidence width: An objective measure for association rule novelty. In *Workshop on quality issues, measures of interestingness and evaluation of data mining models QIMIE* (Vol. 9).

Bayardo, R., Agrawal, R., Gunopulos, D. (2000). Constraint-based rule mining in large, dense databases. *Data Mining and Knowledge Discovery, 4*(2), 217–240.

Bayardo Jr., R., & Agrawal, R. (1999). Mining the most interesting rules. In *Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 145–154). ACM.

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

Freitas, A. (1999). On rule interestingness measures. *Knowledge-Based Systems, 12*(5–6), 309–315.

Fukuda, T., Morimoto, Y., Morishita, S., Tokuyama, T. (1996). Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. In *ACM SIGMOD Record* (Vol. 25, pp. 13–23). ACM.

Gebhardt, F. (1991). Choosing among competing generalizations. *Knowledge Acquisition, 3*(4), 361–380.

Goodman, L., & Kruskal, W. (1954). Measures of association for cross classifications. *Journal of the American Statistical Association, 49*(268), 732–764.

Hills, J., Davis, L.M., Bagnall, A. (2012). Interestingness measures for fixed consequent rules. In *Intelligent Data Engineering and Automated Learning - IDEAL 2012* (pp. 68–75).

Hussain, F., Liu, H., Suzuki, E., Lu, H. (2000). Exception rule mining with a relative interestingness measure. In *Knowledge discovery and data Mining. Current issues and new applications* (pp. 86–97).

de la Iglesia, B., Richards, G., Philpott, M., Rayward-Smith, V. (2006). The application and effectiveness of a multi-objective metaheuristic algorithm for partial classification. *European Journal of Operational Research, 169*(3), 898–917.

Lavrač, N., Flach, P., Zupan, B. (1999). Rule evaluation measures: A unifying view. In *Inductive logic programming* (pp. 174–185).

Liu, H., Lu, H., Feng, L., Hussain, F. (1999). Efficient search of reliable exceptions. In *Methodologies for knowledge discovery and data mining* (pp 194–204).

Liu, H., Liu, L., Zhang, H. (2011). A fast pruning redundant rule method using Galois connection. *Applied Soft Computing, 11*(1), 130–137.

Major, J., & Mangano, J. (1995). Selecting among rules induced from a hurricane database. *Journal of Intelligent Information Systems, 4*(1), 39–52.

Ohsaki, M., Kitaguchi, S., Okamoto, K., Yokoi, H., Yamaguchi, T. (2004). Evaluation of rule interestingness measures with a clinical dataset on hepatitis. In *Knowledge discovery in databases: PKDD 2004* (pp. 362–373).

Paper Authors (2012). Companion website. https://sites.google.com/site/brutesuppression. Accessed 30 Nov 2012.

Reynolds, A., & de la Iglesia, B. (2006). Rule induction using multi-objective metaheuristics: Encouraging rule diversity. In *International joint conference on neural networks, 2006. IJCNN'06.* (pp 3343–3350). IEEE.

Richards, G., & Rayward-Smith, V. (2001). Discovery of association rules in tabular data. In *Proceedings of the IEEE international conference on data mining* (p. 465). IEEE Computer Society.

Richards, G., & Rayward-Smith, V. (2005). The discovery of association rules from tabular databases comprising nominal and ordinal attributes. *Intelligent Data Analysis, 9*(3), 289–307.

Sarma, P.K.D., & Mahanta, A.K. (2012). Reduction of number of association rules with inter itemset distance in transaction databases. *International Journal of Database Management Systems, 4*(5), 61–82.

Shaharanee, I.N.M. and Hadzic, F., Dillon, T.S. (2011). Interestingness measures for association rules based on statistical validity. *Knowledge-Based Systems, 24*(3), 386–392.

Tamir, R., & Singer, Y. (2006). On a confidence gain measure for association rule discovery and scoring. *The VLDB Journal, 15*(1), 40–52.

Xu, Y., Li, Y., Shaw, G. (2011). Reliable representations for association rules. *Data & Knowledge Engineering, 70*(6), 555–575.