Better Rulesets by Removing Redundant Specialisations and Generalisations in Association Rule Mining

Henry Petersen
University of Sydney
hpetersen@sydney.edu.au

Josiah Poon
University of Sydney

Simon Poon
University of Sydney

Clement Loy
University of Sydney

Abstract

Association rule mining is a fundamental task in many data mining and analysis applications, both for knowledge extraction and as part of other processes (for example, building associative classifiers). It is well known that the number of associations identified by many association rule mining algorithms can be so large as to present a barrier to their interpretability and practical use. A typical solution to this problem involves removing redundant rules. This paper proposes a novel definition of redundancy, which is used to identify only the most interesting associations. Compared to existing redundancy based approaches, our method is both more robust to noise, and produces fewer overall rules for a given data (improving clarity). A rule can be considered redundant if the knowledge it describes is already contained in other rules. Given an association rule, most existing approaches consider rules to be redundant if they add additional variables without increasing quality according to some measure of interestingness. We claim that complex interactions between variables can confound many interestingness measures. This can lead to existing approaches being overly aggressive in removing redundant associations. Most existing approaches also fail to take into account situations where more general rules (those with fewer attributes) can be considered redundant with respect to their specialisations. We examine this problem and provide concrete examples of such errors using artificial data. An alternate definition of redundancy that addresses these issues is proposed. Our approach is shown to identify interesting associations missed by comparable methods on multiple real and synthetic data. When combined with the removal of redundant generalisations, our approach is often able to generate smaller overall rule sets, while leaving average rule quality unaffected or slightly improved.

Keywords: Association Rule Mining;Redundancy

1 Introduction

Association rule mining is an important task in knowledge extraction and data analysis. Formally, let A = $\{a_1, a_2, \dots, a_M\}$be a set of M attributes. We then define N data D = $\{d_1, d_2, \dots, d_N\}$ where each data d_i contains a subset of the attributes in A (i.e. $d_i \subseteq A, \forall d_i \in D$). The association rule mining task seeks to find all *interesting* rules of the form $X \Rightarrow Y$, where X and Y are disjoint subsets of A.

It is well known that the number of rules generated can be so large as to obscure their interpretation, presenting a barrier to practical use (Zaki 2000). Ideally, we wish to find only the most interesting rules. This can be broken into two tasks; identification of truly interesting rules, and the removal of those which are simply redundant artefacts of other rules. A rule is considered interesting if it produces a value superior to some predetermined threshold according to a chosen function of interestingness M(·) (e.g. Support, confidence, Fishers P). Standard thresholds exist for some measures of interestingness (e.g. $P < 0.05$ for measures of

statistical significance), while other for other measures the threshold is determined in a more ad hoc manor. Our work focuses primarily on statistical measures of interestingness.

When comparing two rules $X \Rightarrow Z$ and $XQ \Rightarrow Z$ to evaluate redundancy, existing approaches ignore data containing only part of the antecedent. As a result, non-interesting rules can be retained, and potentially interesting rules discarded. This paper proposes an alternate definition of redundancy (which we call robust redundancy) that utilises such information to improve the quality of the discovered rules.

Examples are given demonstrating the ability of robust redundancy to correctly identify interesting rules which would otherwise have been missed by more classical approaches. An algorithm for generating rules with robust redundancy is also proposed and evaluated on both real and artificial data.

Within the literature, there are several sub-problems that have been studied. Examples include rules with fixed (Verhein and Chawla 2007), or single attribute (Hämäläinen 2012) consequents, numerical data (Song and Ge 2013), or negative associations (Hämäläinen 2012, Li and Zaiane 2015) (e.g. rules of the form $X \Rightarrow \neg Z$). In this paper we focus on the problem of positive rules from binary data with single attribute consequents.

In this paper we refer to the concepts of rule *generalisations* and *specialisations*. Given two rules $X \Rightarrow Z$ and $Y \Rightarrow Z$ (where Z is a single attribute and X and Y are sets of attributes), the rule $Y \Rightarrow Z$ is considered to be a *generalisation* of $X \Rightarrow Z$ if Y is a proper subset of X. Similarly, rule $Y \Rightarrow Z$ is a *specialisation* of $X \Rightarrow Z$ if Y is a proper superset of X.

2 Background

A key problem for association rule mining is measuring how interesting a rule is. Traditionally this is done using *support* and *confidence* (analogous to the sample probability of a rule and the conditional probability of Y given X respectively). Many alternate approaches for measuring interestingness have been proposed (Piatetsky-Shapiro 1991, Brin, Motwani et al. 1997), several of which are presented in Table 1. The interested reader is directed to the literature for further information (Tan, Kumar et al. 2004). The experimental work in this paper focuses on statistical measures interestingness, where rule interestingness is analogous to statistical significance.

The size of the search space is a major concern when mining association rules. Prior work often employs heuristics such as maximum rule lengths, fixed consequents, or frequency thresholds in order to control this (Hämäläinen 2010). To our knowledge, only Hämäläinen's Kingfisher algorithm is able to identify all significant rules using the current definition of non-redundancy. We extend the Kingfisher approach for rule generation in section 4.

It is well established that the number of rules identified can often be so large as to hamper their interpretation (Aggarwal and Yu 2001). The concept of *redundancy* can be used to control the number of rules. Consider a hypothetical study of supermarket transactions which identifies that people who buy a soft drink will also buy chips. Further analysis may also identify that people who buy soft drink on Tuesday will buy chips. However the condition that it be Tuesday does not improve the quality of the association. The rule likely exists because the association between people buying soft drink and chips holds regardless of whether it is Tuesday or not. The association between people buying soft drink on Tuesday and buying chips is *redundant*.

Measure	**Formulae**
Support (Agrawal, Imieli et al. 1993)	$\frac{m(X)}{N}$
Confidence (Agrawal, Imieli et al. 1993)	$\frac{m(XY)}{N}$
Interest (Brin, Motwani et al. 1997)	$\frac{N \times m(XY)}{m(X) \times m(Y)}$
Leverage (Piatetsky-Shapiro 1991)	$\frac{m(XY)}{N} - \frac{m(X)m(Y)}{N^2}$
X^2	$\frac{N^5 - Leverage(X \Rightarrow Y)^2}{m(X)m(\neg X)m(Y)m(\neg Y)}$
Fisher's P	$\sum_{i=0}^{\min(m(X\neg Y),m(Y\neg X))} \frac{\binom{m(X)}{m(XY)+i}\binom{m(\neg X)}{m(\neg X\neg Y)+i}}{\binom{N}{m(Y)}}$

Table 1: Several common interestingness measures for a rule $X \Rightarrow Y$ expressed in terms of partial frequency counts. N is the size of the data, and $m(\cdot)$ is the frequency function.

Several authors have proposed formal means for defining redundant rules (Zaki 2000, Aggarwal and Yu 2001, Ashrafi, Taniar et al. 2004, Webb 2006, Webb 2007). A definition of redundancy suitable for use with a general goodness measure (assuming single attribute consequents) was first proposed in 2010 by Hämäläinen (Hämäläinen 2010). Hämäläinen defines a rule R to be redundant if some more general rule (i.e. a rule whose antecedent is a subset of the antecedent of R) has equal or better utility with respect to some goodness measure. We repeat this more formally in Definition 1.

> *Definition 1: Classical Redundancy*
>
> *Consider two rules $X \Rightarrow Z$ and $XQ \Rightarrow Z$ where X and Q are disjoint sets of items, and Z is a single item of value a. Let $M(\cdot)$ be an increasing measure of rule interestingness. Rule $XQ \Rightarrow Z$ is redundant with respect to rule $X \Rightarrow Z$ if $M(XQ \Rightarrow Z) \leq M(X \Rightarrow Z)$.*

We note that when comparing rules based on some arbitrary goodness measure, complications can arise due to complex interactions between constituent attributes. That such interactions can give rise to spurious associations has been studied (McGrane and Poon 2010, Webb 2010), however less attention has been paid to how it might obscure useful relationships.

Models of redundancy that remove spurious generalisations is a problem that has seen limited interest within the community. Removing spurious generalisations was first looked at in 2001 by Liu et al. in their work on non-actionable rules (Liu et al. 2001). For a rule r0 and the set of its decedents R = $\{r_1, r_2, \ldots, r_N\}$, they define a rule r0 to be *non-actionable* if it is not interesting over the domain where instances matching at least one antecedent in R are removed. Essentially, they claim a rule must cover some unique set of instances (with respect to the set of its specialisations) in which the relationship described still holds. A rule has no utility with respect to the set of its specialisations if it does not cover such a set of instances.

A similar concept to non-actionable rules was proposed by Webb in his work on self-sufficient itemsets (Webb 2010). This work builds upon the concept of an *exclusive domain* for a given itemset. Formally, given an itemset s and its specialisations S, the exclusive domain of s is defined to the domain of s minus the union of the domains of all itemsets in S. After generalising the concepts of redundancy and productivity (Webb 2006, Webb 2007) for use with itemsets, an itemset is defined to be self-sufficient if it is productive and non-redundant both with respect to the entire data and its exclusive domain.

In many respects, self-sufficient itemsets can be considered an extension of non-actionable rules for use in an itemset context. However, it is noteworthy that itemsets must also be productive and non-redundant. This pruning of both specialised and general itemsets is similar to our work with robust redundancy described in this paper, although it is performed in an itemset context. We also examine how to avoid pruning specialisations where redundancy is likely to be an artefact of interactions between constituent attributes.

3 Robust Redundancy

A rule is considered redundant when it adds no information over another rule. We claim that classical redundancy makes such a comparison using incomplete information.

Depending on the interestingness measure M(·) in use, M(X ⇒ Z) is computed using the frequencies XZ, ¬XZ, X¬Z, and ¬X¬Z. Note that directly comparing rules M(X ⇒ Z) and M(XQ ⇒ Z) does not consider transactions including only part of the rule antecedent (i.e. frequencies of X¬QZ, ¬XQZ, X¬Q¬Z, and ¬XQ¬Z).

Association rule mining can be confounded by noise and complex relationships between variables. Potential lack of control over the data collection process can further complicate matters. Such noise could artificially raise or lower the measured interestingness value of a rule, which could lead to interesting rules being incorrectly excluded.

We propose using additional information in an attempt to avoid excluding interesting rules. We also seek to identify seemingly interesting rules that are simply artefacts of groups of their specialisations. We refer to these approaches as specialisation and generalisation redundancy respectively.

3.1 Specialisation Redundancy

We propose an alternate approach to redundancy in Definition 2. We augment the classical approach given in Definition 1 by not eliminating a rule XQ ⇒ Z if the partial frequencies can be used to demonstrate the attributes in Q add value. This is accomplished by computing the strength of the association between X and Z conditioned on Q, and comparing it against the strength of the marginal association. If the conditional association between X and Z improves over the strength of the previous association, we obtain evidence that the addition of Q adds value to the existing rule.

> *Definition 2: Specialisation Redundancy*
>
> *Consider two rules X ⇒ Z and XQ ⇒ Z where X and Q are disjoint sets of attributes, and Z is a single attribute. Let M(·) be an increasing measure of rule interestingness. Rule XQ ⇒ Z is specialisation redundant with respect to rule X ⇒ Z if M(XQ ⇒ Z) ≤ M(X ⇒ Z), and M(X ⇒ Z | Q) ≤ M(X ⇒ Z).*

Computing the conditional association requires frequencies for ¬XQZ and ¬XQ¬Z (in addition to the frequencies XQZ and XQ¬Z). We do not consider the association between X and Z conditioned on ¬Q, as the rule we are seeking to obtain evidence for is XQ ⇒ Z, which contains Q.

3.1.1 Example

Consider hypothetical data sampling 3 binary variables X, Y, and Z. Assume 1000 data points with the probabilities expressed in Table 2a. From these probabilities, observe that a strong

dependency exists between Z and the itemset XY. We now examine the rules X ⇒ Z, Y ⇒ Z, and XY ⇒ Z as we vary the joint probability of variables X and Y.

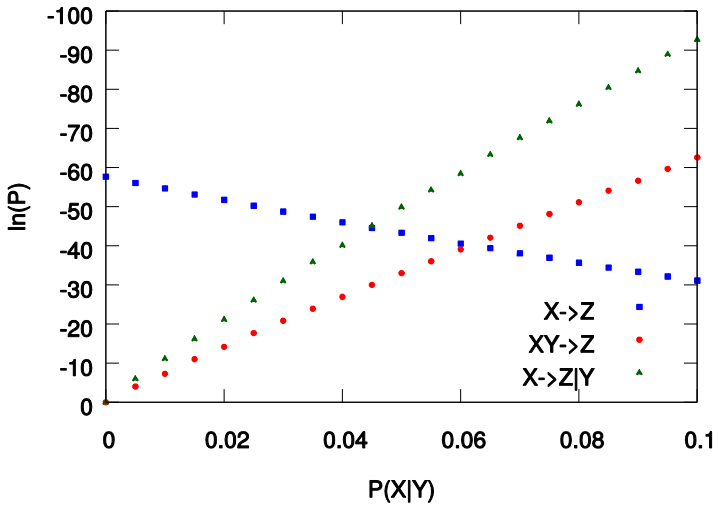


Figure 1: ln(P-values) for rules X ⇒ Z, Y ⇒ Z, and XY ⇒ Z versus conditional probability of X and Y.

Figure 1 plots rule quality against the conditional probability of X given Y. For larger conditional probabilities we can observe that the quality of the rule XY ⇒ Z is superior to that of rule X ⇒ Z. As the overlap between data containing X and data containing Y decreases, the quality of the more general rule X ⇒ Z surpasses its specialisation. Consequently the rule XY ⇒ Z is removed as redundant, obscuring the true underlying structure of the data.

(a)

P(X)	0.3
P(Y)	0.3
P(Z|X,Y)	0.8
P(Z|X,¬Y)	0.4
P(Z|¬X,Y)	0.4
P(Z|¬X,¬Y)	0.6

(b)

(X)	0.5
P(Y)	0.5
P(Z|X,Y)	0.5
P(Z|X,¬Y)	0.1
P(Z|¬X,Y)	0.1
P(Z|¬X,¬Y)	0.1

Table 2: Marginal and conditional probabilities for several combinations of variables used in the motivating examples for robust redundancy.

Holding the marginal probabilities constant, the frequencies of both X and Y decrease along with the conditional probability of X given Y. In order to support the rule XY ⇒ Z, we require data containing both (or neither) XY and Z. Hence, as the number of data with XY and Z decreases, so too does the evidence available to evaluate it. That the general rule X ⇒ Z surpasses the true rule XY ⇒ Z in quality as the conditional probability decreases is a reflection of this.

By comparing rules X ⇒ Z and XY ⇒ Z using robust redundancy (i.e. including the conditional dependencies) we make more effective use of available data to evaluate the rules. In the example given in Figure 1, rule XY ⇒ Z is retained as non-redundant for conditional probabilities greater than ~0.045. This is in contrast to classical redundancy, where the threshold for retaining XY ⇒ Z is ~0.062. Although in both cases the conditional probability of X given Y eventually reaches a point where insufficient evidence for the specialised rule exists, the range of values for which robust redundancy can still retain XY ⇒ Z is increased.

3.2 Generalisation Redundancy

It is possible for general rules to exist that only appear interesting due to the presence of one or more interesting specialisations. Definition 3 outlines a concept we call *Robust Generalisation Redundancy*. A rule $X \Rightarrow Z$ is generalisation redundant if for all non-redundant specialisations $XQ \Rightarrow Z$, the rule $X \Rightarrow Z \mid \neg Q$ is uninteresting (its goodness is less than the required threshold).

If a rule $X\neg Q \Rightarrow Z$ is interesting, we obtain evidence that the generalised rule is interesting even in the absence of the terms in Q. If after identifying all other interesting rules we cannot find evidence that $X \Rightarrow Z$ is interesting in the absence of the additional terms in its specialisations, we consider it redundant. Computing the conditional association on $\neg Q$ uses the frequencies $X\neg QZ$ and $X\neg Q\neg Z$. Hence, by applying both specialisation and generalisation redundancy we consider all frequencies in the sample data.

> *Definition 3: Generalisation Redundancy*
>
> *Consider a rule $X \Rightarrow Z$ and the complete set of its non-redundant specialisations R. Let $M(\cdot)$ be an increasing measure of rule interestingness, and α be the corresponding goodness threshold. Rule $X \Rightarrow Z$ is generalisation redundant with respect to R if $M(X \Rightarrow Z \mid \neg Q) \leq \alpha$ for all rules $XQ \Rightarrow Z$ in R.*

3.2.1 Example

Consider hypothetical data containing drug prescriptions and a corresponding binary patient outcome. Assume there are two drugs (X and Y) which work in combination to produce a positive outcome. Neither drug will produce a positive outcome on its own (a baseline probability for positive outcome of 0.1 is used). The exact probabilities used can be found in Table 2b.

When the conditional probability of X given Y is 1, the measured quality of the rules $X \Rightarrow Z$, $Y \Rightarrow Z$, and $XY \Rightarrow Z$ will be identical and maximal. The quality of these rules decreases with this conditional probability, with the quality of the general rules decreasing at the greatest rate. However despite the underlying structure of the data indicating that neither X or Y alone support a positive outcome, the strength of these associations will likely remain quite high.

When comparing rules $X \Rightarrow Z$ and $XY \Rightarrow Z$, examining the rule $X \Rightarrow Z \mid \neg Y$ (i.e. conditioned on the absence of the additional terms Y) indicates that there is no evidence to support the rule $X \Rightarrow Z$ without also including the features Y. As $XY \Rightarrow Z$ is the only identified specialisation of $X \Rightarrow Z$, and we have no evidence to indicate $X \Rightarrow Z$ is valid without the additional features, we consider it redundant.

Finally, we acknowledge that such an approach could potentially over-fit and remove valid general rules. We address this concern in the following section on redundancy chaining.

3.3 Redundancy Chaining

Classical redundancy as defined in Definition 1 is transitive. If a rule $XQY \Rightarrow Z$ is redundant with respect to a generalisation $XQ \Rightarrow Z$, and $XQ \Rightarrow Z$ is redundant with respect to $X \Rightarrow Z$, then $XQY \Rightarrow Z$ will be redundant with respect to $X \Rightarrow Z$. This result is straightforward to prove.

(a)

Attr	Freq.
ABCD	10
ABD	10
ACD	10
AD	10
BC	30
BD	10
CD	10
D	10

(b)

Rule	ln(P)
A ⇒ D	-19.33
AB ⇒ D	-8.10
ABC ⇒ D	-3.78
A ⇒ D|B	-18.75
A ⇒ D|BC	-20.56
AB ⇒ D|C	-7.83

Table 3: Sample data and rules for lemma 1.

Unfortunately, the same relation does not hold for the proposed robust redundancy. A proof that specialisation redundancy is nontransitive is given in Lemma 1. The fact that specialisation redundancy is non-transitive can lead to some interesting behaviour. Assume a rule r exists that is specialisation redundant with respect to one or more generalisations $r_0, \dots, r_i$. Let $r_0, \dots, r_i$ be redundant with respect to rules $r_{i+1}, \dots, r_n$. Despite being a redundant specialisation of other rules, r is non-redundant with respect to all non-redundant generalisations. We take the view that in such a situation, the rule r should be considered non-redundant.

Lemma 2: Using redundant rules when evaluating generalisation redundancy allows for additional rules to be included.

Consider three rules A ⇒ D, AB ⇒ D, and ABC ⇒ D generated from the data in Table 4 using the confidence measure with a threshold of 0.6.

According to Definition 3 and the confidence scores for the above rules, AB ⇒ D | ¬C is uninteresting so AB ⇒ D is redundant w.r.t. ABC ⇒ D. When evaluating generalisation redundancy without redundant rules, the uninteresting rule A ⇒ D | ¬(BC) implied A ⇒ D is redundant. When evaluating generalisation redundancy with redundant rules, as A ⇒ D | ¬B is interesting A ⇒ D is non-redundant. ■

We also prove that whether or not redundant attributes are counted effects generalisation redundancy in Lemma 2. As above, we elect not to allow redundant rules to influence the redundancy of another rule. In contrast to specialisation redundancy, this will lead to the exclusion of additional rules (comparing against additional rules raises the chance of inclusion as generalisation redundancy requires a rule be uninteresting with respect to ALL its specialisations).

Lemma 1: Specialisation redundancy is nontransitive.

Let A ⇒ D, AB ⇒ D, and ABC ⇒ D be three rules generated from data in Table 3 using the log of Fishers P.

As the interestingness of the rules AB ⇒ D and A ⇒ D | B is worse than that of the rule A ⇒ D, AB ⇒ D is redundant w.r.t. A ⇒ D.

As the interestingness of the rules ABC ⇒ D and AB ⇒ D | C is worse than that of the rule AB ⇒ D, ABC ⇒ D is redundant w.r.t. AB ⇒ D.

As the interestingness of the rule A ⇒ D | BC is better than that of the rule A ⇒ D, the rule ABC ⇒ D is non-redundant w.r.t. A ⇒ D. ■

Not using redundant rules to support retaining otherwise redundant generalisations can produce the following interesting situation. Assume a rule r_1:Y ⇒ A where *conf*(Y ⇒ A) = 1 and *supp*(Y)= *supp*(A). Then for all rules of the form r_1 X ⇒ A where Y=XQ (i.e. generalisations of Y ⇒ A), the frequency of X¬QA will be 0, and the rule X¬Q ⇒ A will be uninteresting. By Definition 3, r_1 is the only possible non-redundant rule with consequent A.

While it may in fact be desirable to keep such a rule, care must be taken to avoid confounding caused by the addition of independent attributes. We demonstrate how such confounding might occur by providing an extension of the above example. Consider the rule YZ ⇒ A for some variable Z where *supp*(ZA) = 1. It is simple to see that *conf*(YZ ⇒ A) = 1, *supp*(YZ)=*supp*(A), and *freq*(Y⌐ZA) = 0. The rule Y ⇒ A will be considered redundant.

(a)

Attr	Freq.
ABCD	60
AB	20
ACD	10
AD	10

(b)

Rule	Conf
A ⇒ D	0.90
AB ⇒ D	0.75
ABC ⇒ D	1.00
A ⇒ D|⌐B	1.00
A ⇒ D|⌐B⌐C	0.50
AB ⇒ D|⌐C	0.00

Table 4: Sample data and rules for lemma 2.

By Occam's Razor we prefer a more general rule over its specialisations unless evidence can be obtained to suggest otherwise. Using only generalisation redundancy can violate this principle as no evidence is ever considered to support YZ ⇒ A over Y ⇒ A. In the worst case, for a given consequent only one highly specific rule will be selected with all others being made redundant. Therefore specialisation (or classical) redundancy should usually be employed before generalisation redundancy. We note however that in some cases (such as those where we prefer to generate more specific rules), generalisation redundancy may be applied first.

4 Rule Mining Algorithm

Pseudocode for our algorithm (an extension of the Kingfisher algorithm (Hämäläinen 2010, Hämäläinen 2012) is given in Algorithm 1. We find non-redundant rules using the natural log of Fisher's P value (a decreasing measure) in a three stage process:

1. All potentially non-redundant rules with some minimum log P-value are identified.
2. Rules identified in stage 1 are examined and specialisation redundant rules are pruned.
3. Remaining rules are examined and generalisation redundant rules are pruned.

Stage 1 is a BFS over itemsets, which is described in Algorithm 1. It is equivalent to the Kingfisher algorithm, however uses less strict pruning to avoid removing potentially relevant rules prior to stages 2 and 3 (see section 4.1).

Algorithm 1: Search for potentially non-redundant rules

Input: Set of attributes A, Dataset D, Decreasing goodness measure M(), Threshold α
Output: Set of potentially non-redundant rules R

```
1.  sort A decreasing by frequency in D
2.  k = 1
3.  while k ≤ |A|
4.    for each candidate attribute k-set C
5.      for each rule r: C\{a} ⇒ a ∀ a ∈ C
6.        if M(r) ≤ α
7.          add r to result set R
8.          C.pbest[a] = min(C.pbest[a], M(r))
9.      for each rule r: C\{a} ⇒ a ∀ a ∈ A
10.       bnd = lower bound on M for all specialisations of r
11.       if bnd ≥ α
12.         C.possible[a] = false
13.       if a ∈ C
14.         bndonq = lower bound on M for all specialisations of r conditioned on the
                       additional attributes
15.         if bnd > C.pbest[a] and bndonq > C.pbest[a]
16.           C.possible[a] =false
17.     if C.possible[a] == false for all a ∈ A
18.       Remove all attribute sets containing C from candidate attribute sets
19.   k = k + 1
20. return R
```

4.1 Identification of Potentially Non-Redundant Rules

The search begins at step 1 by sorting attributes in decreasing order of frequency. Any attribute whose frequency is too low to produce an interestingness value greater than α is removed at this stage. Level 1 search nodes are then created for each remaining attribute (a node is called level k if it represents a set of k attributes). Steps 2 – 4 describe the breadth first search over nodes. At each level i we create nodes by taking the union of two i-1 nodes.

At steps 5-7, for each level k node X corresponding to attributes $x_1, x_2, \ldots, x_k$, the P-values of the k rules $X \backslash x_i \Rightarrow x_i$ are computed and compared against the minimum threshold α. The frequency of set X is calculated and stored, with P-values for rules being computed using frequencies of parent nodes. The number of iterations over the dataset is therefore limited to the number of nodes considered.

Each node maintains a length |A| bit vector of possible consequents (attributes A where the rule XQ ⇒ A is possible). These vectors are initialised using bitwise and of vectors for parent nodes. As a node is processed, lower bounds on the log Fisher's P value are computed for all rules of the form XQ\A⇒A for all A in A. If bounds for all attributes exceed the relevance threshold (the vector of possible consequents is 0), the node is pruned from the search. The bounds used were first reported by Hämäläinen (Hämäläinen 2010), and are reproduced in Table 5. This is described in steps 9-12, 17, and 18.

Each node X contains a vector with the best previous P-value for rules with consequent xi in X. This is computed in step 8. Similar to the possible bit vector, these vectors are merged from parents when the node X is created. Using classical redundancy, if the bound on P-values for rules with a given consequent exceeds the corresponding value in this vector that consequent

can also be considered impossible. Using robust redundancy (see Definition 2), we also test that the bound on rule XQ ⇒ A|Q is worse than the previous best value.

The Fishers P-value for rule XQ ⇒ A|Q takes its smallest value when the number of instances containing sets QA¬X and QX¬A are 0 and QXA and Q¬X¬A are as large as possible. This occurs when freq(QXA) = freq(XA) and freq(Q¬X¬A) = freq(¬X¬A). We therefore compute the bound for XQ ⇒ A|Q using bnd3 from Table 5 with parameters f(XA)=freq(XA), f(X)=freq(XA), f(A)=freq(XA), and N=freq(XA)+freq(¬X¬A). This computation is performed in steps 13-14.

Algorithm 2: Prune redundant specialisations

Input: Set of rules R, Dataset D, Decreasing goodness measure M()
Output: Set of non (specialisation) redundant rules R

1. for each consequent C
2. R_C = all rules with consequent C
3. sort R_C increasing on the length of the antecedent
4. for each rule r_i in R_C
5. for each rule r_k in $R_C[i+1, |R_C|]$
6. X = antecedent(r_i)
7. Y = antecedent(r_j)
8. if $X \subset Y$
9. Q = Y\X
10. if $M(X \Rightarrow C) \leq M(Y \Rightarrow C)$ and $M(X \Rightarrow C) \leq M(X \Rightarrow C \mid Q)$
11. delete r_k
12. return R

Algorithm 3: Prune redundant generalisations

Input: Set of rules R, Dataset D, Decreasing goodness measure M()
Output: Set of non-redundant rules R

1. for all r in R
2. keep(r) = false
3. hasspec(r) = false
4. for each consequent C
5. R_C = all rules with consequent C
6. sort R_C increasing on the length of the antecedent
7. for each rule r_i in R_C (reverse order)
8. if keep(r_i) or not hasspec(r_i)
9. for each rule r_k in $R_C[i+1, |R_C|]$
10. Y = antecedent(r_i)
11. X = antecedent(r_j)
12. if $X \subset Y$
13. hasspec(r_i) = true
14. Q = Y\X
15. if $M(X \Rightarrow C \mid \neg Q) \leq \alpha$
16. keep(r_k) = true
17. for all r in R
18. if not keep(r) and hasspec(r)
19. delete r
20. return R

4.2 Pruning the Search Space

The Kingfisher algorithm (Hämäläinen 2010) employs two additional pruning steps to control the size of the search space. The first, referred to as the *lapis philosophorum* principle, deals with the case where all rules of the form XQ ⇒ A become impossible at a given node X{A}. In such a case, A is also an impossible consequence for children of the parent node X, and its possible consequents vector can be updated. This principle is also applied in our approach.

The latter pruning step is pruning based on *minimality*. A rule X ⇒ A is considered minimal iff P(A|X)=1. For a given minimal rule X ⇒ A, any rule of the form XQ ⇒ A or XQA ⇒ B will be either classically redundant or not significant (Hämäläinen 2012).

$\boldsymbol{bnd1}(\lvert\boldsymbol{A}\rvert, \boldsymbol{N}) = \dfrac{\boldsymbol{f}(\boldsymbol{A})!\,\boldsymbol{f}(\neg\boldsymbol{A})!}{\boldsymbol{N}!}$
$bnd2(\lvert X\rvert, \lvert A\rvert, N) = \dfrac{f(\neg X)!\,f(A)!}{N!\,\big(f(A) - f(X)\big)!}$
$bnd3(\lvert XA\rvert, \lvert X\rvert, \lvert A\rvert, N) = \dfrac{f(A)!\,f(\neg A)!\,\big(N - f(XA)\big)!}{N!\,f(\neg A)!\,f(A \neg X)!}$

Table 5: Lower bounds for Fishers P. The function f(·) returns the frequency of its argument in D.

Pruning based on minimality cannot be employed when searching for rules with robust redundancy. We now prove that with robust redundancy it is possible for a specialisation of a minimal rule to be both significant and non-redundant.

Lemma 3: Given data D, an increasing statistical goodness measure M(·), and rule X ⇒ A such that P(A|X)=1, XQ ⇒ A may exist such that M(X ⇒ A) < M(X ⇒ A|Q).

X ⇒ A is minimal implies the frequency of set X¬A is 0. The frequencies of sets XA, ¬XA, and ¬X¬A are unknown.

Let Q be a set of attributes whose corresponding rows in D exactly match the sets XA and ¬X¬A. M(X ⇒ A) increases with each occurrence of XA and ¬X¬A, and decreases with each occurrence of ¬XA. It is easy to observe tat freq(XA)=freq(XQA), freq(¬X¬A)=freq(¬XQ¬A), and freq(¬XA) ≥ freq(¬XQA). Assuming D contains at least one occurrence of ¬XA, M(X ⇒ A|Q) will therefore be greater than M(X ⇒ A). ■

Lemma 4: Given data D, an increasing statistical goodness measure M(·), and a rule X ⇒ A such that P(A|X)=1, there may exist a rule XQA ⇒ B such that M(XA ⇒ B) < M(XA ⇒ B|Q).

X ⇒ A is minimal implies that the frequency of the set X¬A is 0. Hence freq(XA) ≥ freq(XQA). The frequencies of the sets XA, ¬XA, and ¬X¬A are unknown.

Let Q be a set of attributes whose rows in D exactly match the sets CB and ¬C¬B where C=XA. Observe that freq(C)=freq(CQ), freq(¬C¬B)=freq(¬CQ¬B), and freq(¬C) ≥ freq(¬CQ). Assuming D contains at least one occurrence of ¬CB, M(C ⇒ B|Q) will therefore be greater than M(C ⇒ B) (or M(XA ⇒ B|Q) > M(XA ⇒ B)). ■

4.3 Identification of Redundant Rules

The process for pruning specialisation redundant rules is given in Algorithm 2. After grouping rules based on their consequent, rules are sorted in increasing order based on the length of their antecedent (steps 1-3). All pairs of rules within each group are then considered (steps 4-7), and for those pairs which contain a specialisation / generalisation pair we compute the

appropriate M() values to determine specialisation redundancy. Rules which are found to be specialisation redundant are then deleted (steps 8 - 11).

The process for removing redundant generalisations is described in algorithm 3. We begin by grouping rules based on consequent, then sort rules based on antecedent length (lines 4-7). We then traverse the rule list in reverse order, and for each rule consider each of its specialisations in turn (lines 9-12). As a rule only needs to be non-redundant w.r.t. one of its specialisations to be kept, we begin by tagging each rule as an *exclude* (line 2), then changing this value to *true* should an appropriate specialisation be found (lines 14-16). We also tag rules according to whether they have a specialisation to avoid erroneous removal (lines 3 and 13). Rules are then removed according to these tags at the end of the algorithm (lines 17-19).

Finally, as discussed in section 3.3, we need to take care that rules which are already redundant w.r.t. one of their specialisations are not used when determining redundancy for their generalisations. To this end we skip any rule (line 8) which has a specialisation present in the database and is tagged for exclusion (recall that due to the order rules are traversed, such a rule would already have been compared against all its specialisations present in the ruleset).

The running time for stages 2 and 3 are quadratic in the number of rules tested (in general this is dwarfed by the initial search in stage 1). When comparing two rules $X \Rightarrow A$ and $XQ \Rightarrow A$, specialisation redundancy requires the computation of $M(X \Rightarrow A|Q)$, and generalisation redundancy requires $M(X \Rightarrow A|\neg Q)$. For Fisher's P, this requires us to obtain the frequencies for Q, ¬Q, AQ, and A¬Q.

5 Evaluation

Performance is evaluated with respect to three characteristics: total number of rules, overall rule quality, and efficiency. All experiments were run on a PC running Ubuntu Linux, with an Intel I7-4500 processor and 8gb RAM. Performance is also reported for rules generated with the classical definition of redundancy (Definition 1), as well as a baseline with no redundancy based pruning.

5.1 Data

Our evaluation uses the following data covering several domains. Descriptive statistics are also given in Table 6.

- **Mushroom** Mushroom descriptions from the 1981 Audobon Society Field Guide to North American Mushrooms. This data is available in the UCI Machine Learning Repository[1].
- **T10I4D100K** An artificial dataset representing market basket data, obtained from the Frequent Itemset Mining Dataset Repository[2].
- **T40I10D100K** An artificial dataset representing market basket data, obtained from the Frequent Itemset Mining Dataset Repository[3].
- **Diabetes** Collection of real world data reporting traditional Chinese medical herbal prescriptions for diabetes. Includes both the herbs prescribed and a binary classification of the patient outcome as 'good' or 'bad'.
- **Fertility** Collection of real world data reporting traditional Chinese medical herbal prescriptions for fertility. Includes both the herbs prescribed and a binary classification of the patient outcome as 'good' or 'bad'.

[1] https://archive.ics.uci.edu/ml/datasets/Mushroom
[2] http://fimi.ua.ac.be/data/
[3] http://fimi.ua.ac.be/data/

- **Insomnia** Collection reporting traditional Chinese medical herbal prescriptions for insomnia. Includes both the herbs prescribed and a binary classification of the patient outcome as 'good' or 'bad'.
- **Aspergillosis** Text documents (titles and abstracts) for articles considered for inclusion in a systematic review on Aspergillosis (Leeflang, Deeks et al. 2008). Each document is converted to a binary vector indicating the presence or absence of each of 100 words, as well as a binary variable indicating whether the title and abstract was potentially relevant to the review. The words selected were those with the greatest discriminative power when identifying articles relevant to the review.

All values were obtained as the average of 10 independent experiments using a random 50/50 test/training split. Where possible, results are reported with their 95% confidence interval. Statistical significance tests are performed using a P-value of .05. In line with similar work, we measure interestingness using Fishers exact test (we report the natural log of P-values) (Hämäläinen 2012, Li and Zaiane 2015). Thresholds for interesting rules were chosen to strike a balance between permissiveness and execution time, and differ between data and experiments.

Name	# Instances	# Attributes	Avg. Instance Length	Agg. Attribute Freq.
Aspergillosis	4377	101	15.93 ± 0.26	680.51 ± 66.05
Mushroom	8124	119	23.00 ± 0.00	1624.80 ± 358.73
Diabetes	1915	204	10.26 ± 0.11	105.21 ± 30.09
Fertility	766	215	15.73 ± 0.32	59.62 ± 14.21
Insomnia	460	112	13.48 ± 0.25	55.38 ± 11.10
T10I4D100K	100000	870	10.10 ± 0.02	1161.18 ± 74.73
T40I10D100K	100000	942	39.61 ± 0.05	4204.36 ± 249.57

Table 6: Summary of datasets used in the evaluation.

5.2 Size of the Rule Set

Figure 2 shows the number of rules generated for each dataset, pruning approach, and threshold (raw values are given in a table in the appendix). Not shown is the number of rules generated without redundancy based pruning; in all cases these values were substantially (often orders of magnitude) greater than with any type of pruning and were omitted to improve the readability of the graphs. The number of rules generated with robust redundancy is also significantly (P=.05) lower than for classical redundancy with most tested data. For the three exceptions (Insomnia data with thresholds -30 and -35, and Mushroom with threshold -2000), no significant difference in means is observed.

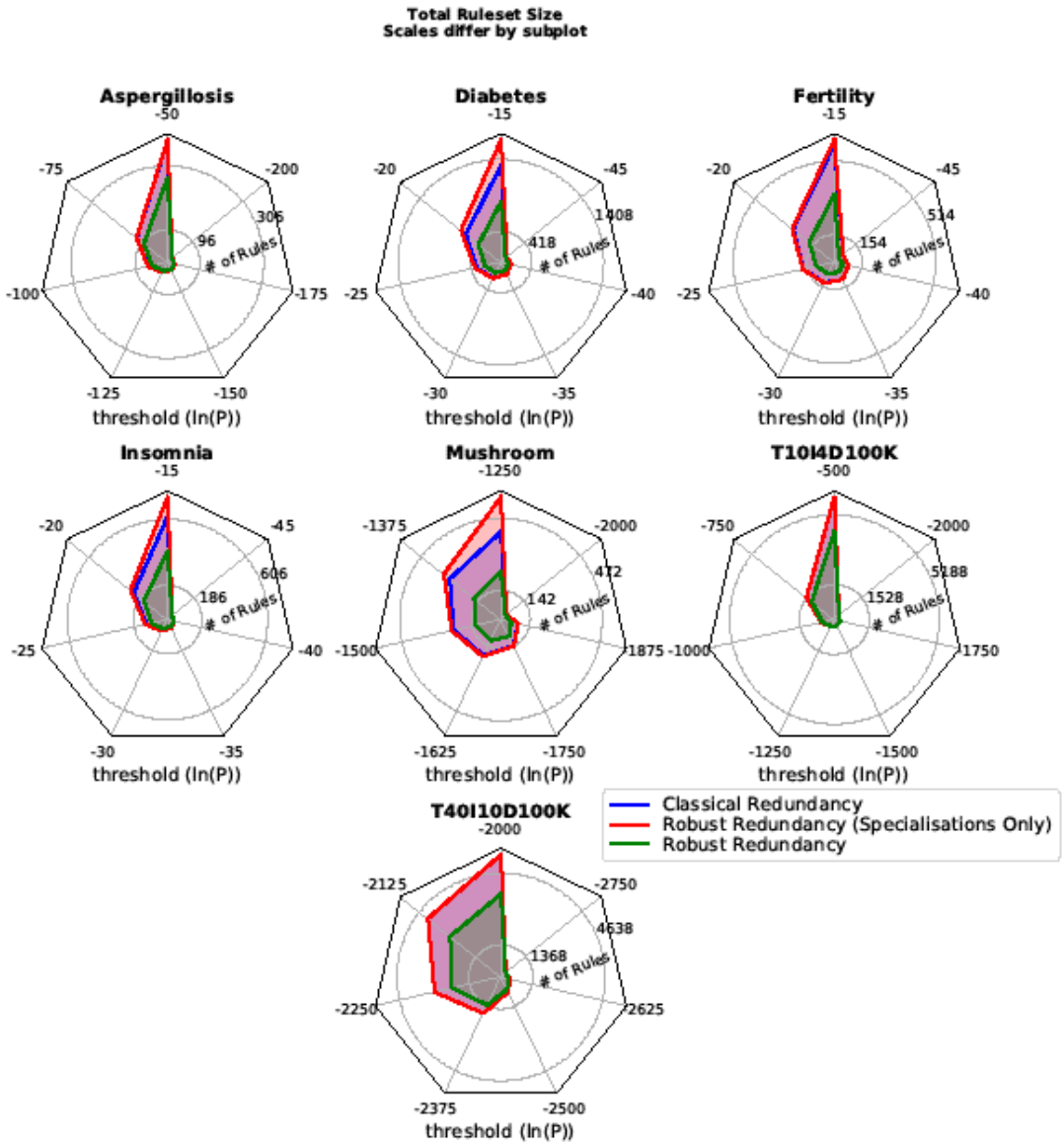


Figure 2: Total ruleset size vs. goodness threshold (ln(P)) with classical, specialisation, and robust (specialisation and generalisation redundancy). Each figure contains one spoke per tested threshold. Values closer to the outside of the plots indicate more rules were produced. Scales differ between subplots.

The cases where no significant difference in the number of non-redundant rules is observed occur using the strictest thresholds. In addition, the difference between the mean number of rules generated appears to increase as the interestingness threshold is relaxed. The number of rules appears to converge as the bound on interesting rules is tightened, and diverge as it is relaxed. This supports the conclusion that our proposed approach is able to produce a practical number of rules from a larger number of potentially interesting associations. This quality is desirable as it allows the use of relaxed interestingness thresholds, lowering the risk of missing potentially useful associations.

We now examine performance when exclusively removing redundant specialisations. Given rule $X \Rightarrow Z$, specialisation redundancy (Definition 2) uses the conditional association $X \Rightarrow Z \mid Q$ to provide an additional chance to obtain evidence for keeping rule $XQ \Rightarrow Z$ (with respect to the classical approach defined in section 2). All specialisations that are not classically

redundant will also not be robust redundant. Robust specialisation redundancy will always return at least as many rules as the classical approach.

5.3 Rule Quality

Next we examine at the quality of the generated rules. As evidence has been given that we should not prune generalisations without first pruning specialisations, no results are reported for pruning generalisations exclusively. Figure 3 shows the mean log P-values for each of the tested redundancy methods and thresholds (raw values are given in the appendix). Despite the smaller generated rule set, it can be seen that in all cases the performance of robust pruning is equivalent or slightly better than for rules generated with classical redundancy.

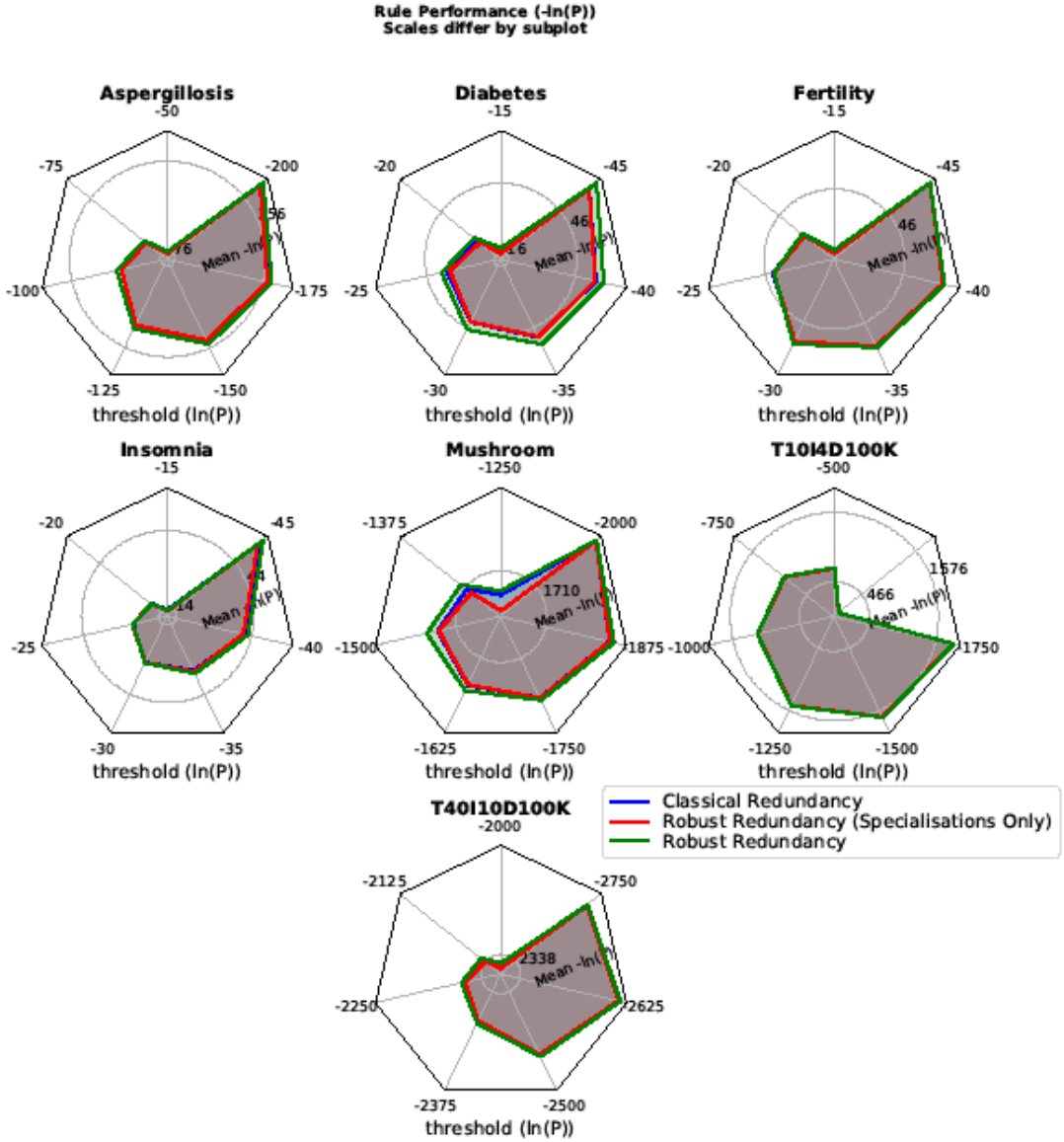


Figure 3: Ruleset performance (average ln(P) vs. goodness threshold (ln(P)) on hold out data with classical, specialisation, and robust (specialisation and generalisation redundancy). Each figure contains one spoke per tested threshold. Values closer to the outside of the plots indicate higher absolute mean quality (lower P values). Scales differ between subplots.

5.4 Efficiency

The expanded search for robust redundancy increases the amount of time and space required. The main factor that effects both computational time and memory requirements is the number of nodes generated during the search. This can be seen by observing the similarity of the trends for the number of nodes generated (Figure 4) against time (Figure 5) and memory (Figure 6) (additional figures displaying these trends are included in the appendix).

Two factors contribute to the increased search space size. As robust specialisation redundancy is more permissive than classical redundancy, the pruning employed during the search must be less aggressive. Additionally, we do not prune based on minimality with robust redundancy. We note however that as our initial search differs from the existing Kingfisher algorithm only in the pruning strategies employed, it maintains the same worst-case time and space complexity.

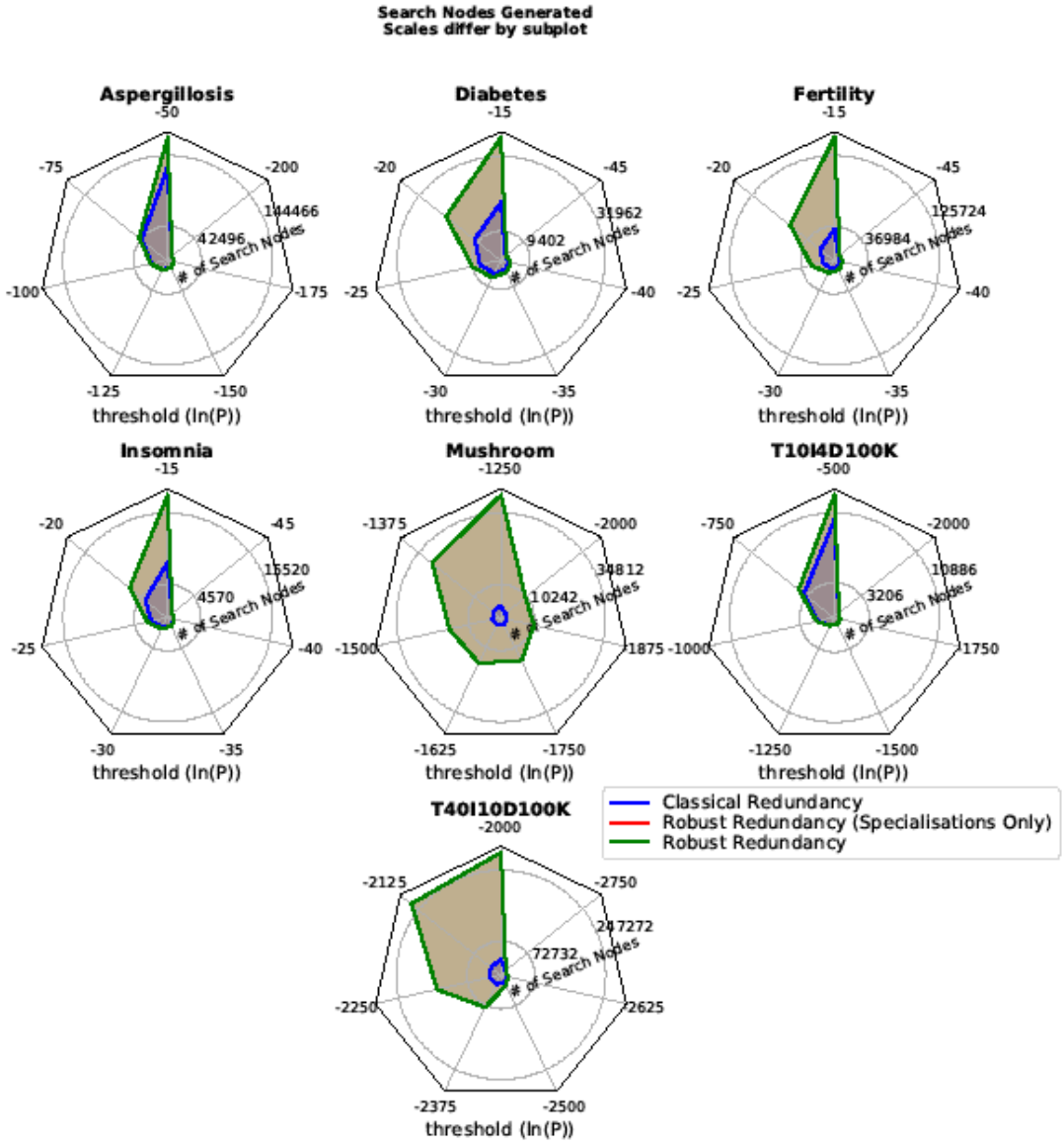


Figure 4: Number of nodes generated during rule search algorithm vs. goodness threshold (ln(P)) with classical, specialisation, and robust (specialisation and generalisation redundancy). Each figure contains one spoke per tested threshold. Values closer to the outside of the plots indicate more search nodes. Scales differ between subplots

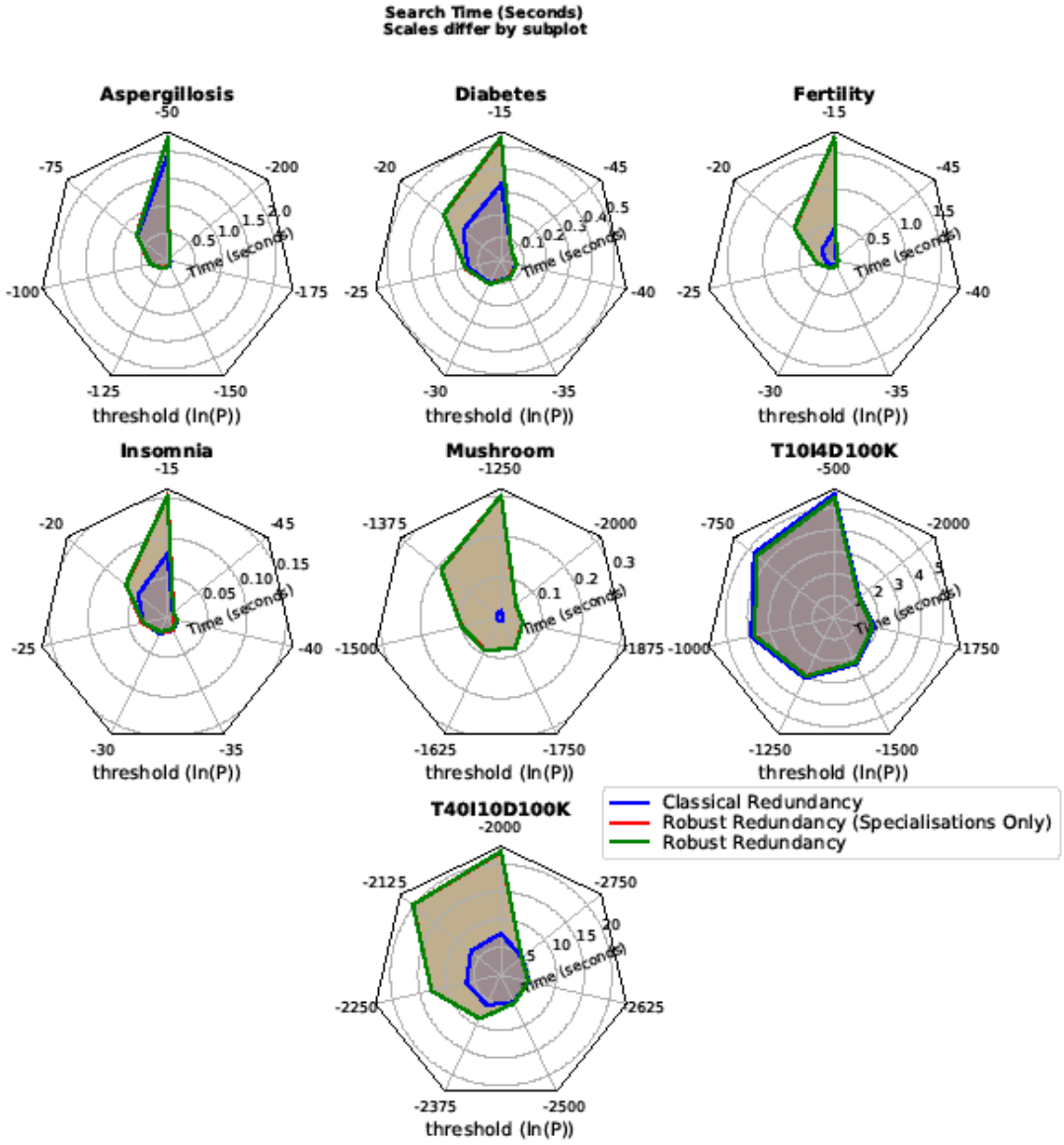


Figure 5: Total search time (seconds) during rule search algorithm vs. goodness threshold (ln(P)) with classical, specialisation, and robust (specialisation and generalisation redundancy). Each figure contains one spoke per tested threshold. Values closer to the outside of the plots indicate higher run times. Scales differ between subplots

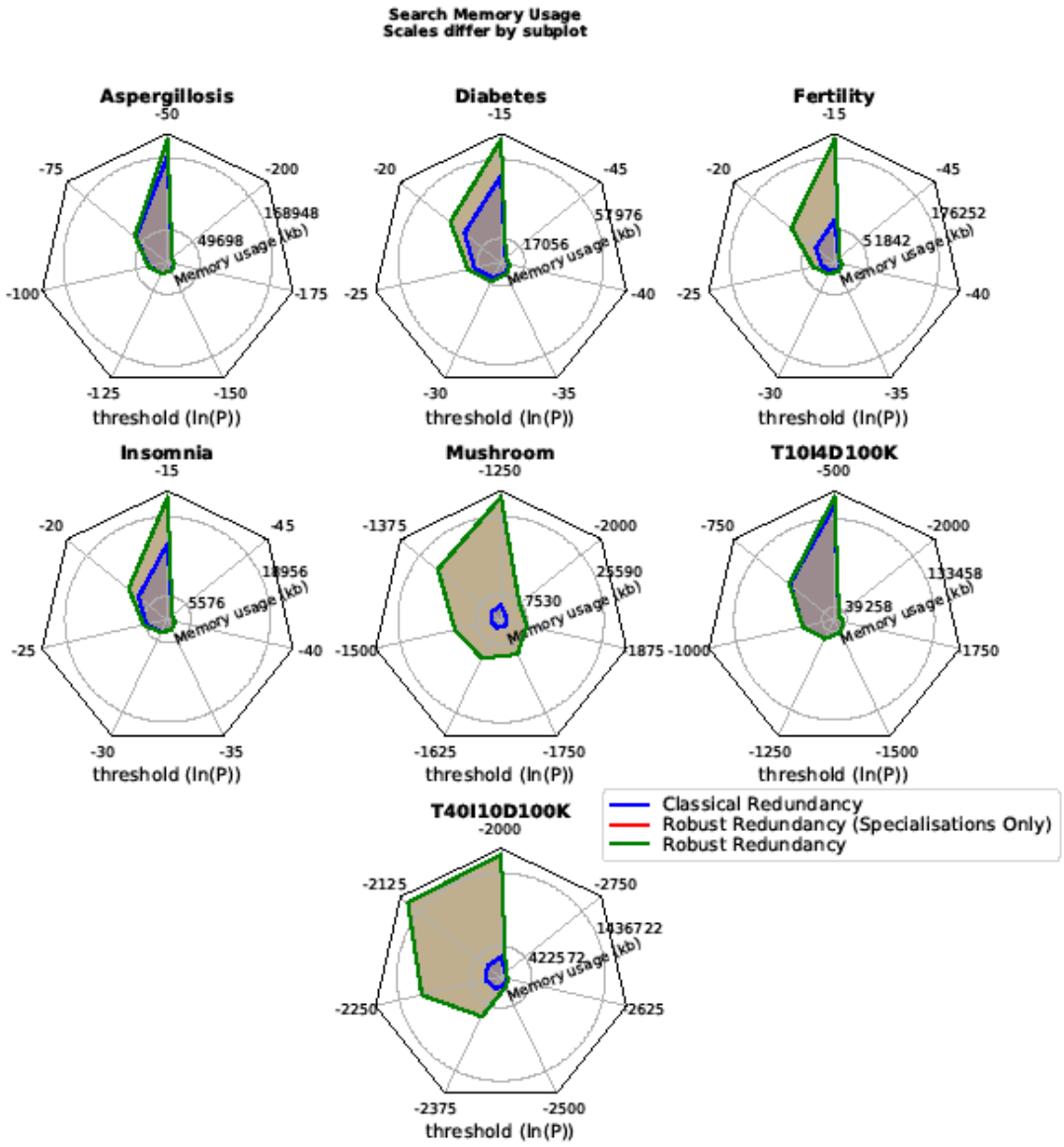


Figure 6: Peak memory usage (kb) during rule search algorithm vs. goodness threshold (ln(P)) with classical, specialisation, and robust (specialisation and generalisation redundancy). Each figure contains one spoke per tested threshold. Values closer to the outside of the plots indicate memory usage. Scales differ between subplots

Figure 2 shows the number of nodes generated when searching with each data (values for robust and specialisation redundancy appear quite similar). Observe the difference in number of nodes generated when searching with robust and classical redundancy varies substantially. Some data (e.g. Aspergillosis and T10I4D100K) differ very little, while the greatest difference is observed for the Mushroom and T40I10D100K data.

In addition to comparing the number of nodes when searching with classical and robust redundancy, it is interesting to examine the number of nodes when no redundancy based pruning is used. A substantial difference exists between the number of nodes generated without pruning when compared to robust redundancy. This implies the bounds computed during the search (reported in Table 5) have a notable effect on the size of the search space.

Three exceptions occur with the Aspergillosis, T10I4D100K, and T40I10D100K data. For Aspergillosis and T10I4D100K we note there is also little difference between the number of nodes generated using robust and classical redundancy. The implication here is that the

majority of the pruning is being done by lapis philosophorum. However, for T40I10D100K there is a substantial difference in performance with robust and classical redundancy.

Figure 5 reports the time for all searches (including pruning in Algorithm 1 and 2). In all cases the required search time was quite manageable. Even for T40I10D100K, all searches completed in less than 30 seconds (all other data completed much quicker). In practice memory appears to become an issue long before time required for the search.

6 Conclusion

This paper considers the problem of identifying and removing redundant associations in association rule mining. A new approach for identifying and removing redundant rules is presented, which we call robust redundancy.

Prior work compared rules based only with their respective contingency tables. Such a comparison fails to consider information included in instances containing only part of the antecedent. Robust redundancy is able to use this information to discover interesting specialisations that would be incorrectly removed with a classical approach.

We also remove rules which are redundant artefacts of their non-redundant specialisations. Unlike previous work (Liu, Hsu et al. 2001, Webb 2010) that evaluate generalisations based on their exclusive domain with respect to the set of all specialisations, we base our method on comparisons to individual rules. We also present the first work using both specialisation and generalisation redundancy in a rule based context (as opposed to the work of Webb (Webb 2010) with itemsets).

We demonstrate several situations in which classical redundancy based approaches can be confounded by interactions between variables, and show that the proposed approach is able to more accurately identify the correct underlying relationships in these situations.

Experimental analysis with multiple real and artificial data demonstrates that robust redundancy often produces smaller overall rule sets compared to classical redundancy. These rule sets hold as well or better in future data than those generated using classical redundancy.

A limitation of our work is the increased time and space requirements of the rule generation compared to classical approaches. We generate rules using a modification of the Kingfisher algorithm with less aggressive pruning; although the worst case complexity does not change, in practice the time requirements can increase substantially. However, as noted in Section 5 performance on experimental data was still good enough to run in reasonable time on standard hardware. Alternate search algorithms and pruning approaches to improve performance in this area are an interesting line of enquiry for future work.

References

Aggarwal, C. C. and P. S. Yu (2001). "A New Approach to Online Generation of Association Rules." *IEEE Trans. on Knowl. and Data Eng*. 13(4): 527-540.

Agrawal, R., T. Imieli, #324, ski and A. Swami (1993). "Mining association rules between sets of items in large databases." *SIGMOD Rec.* 22(2): 207-216.

Ashrafi, M. Z., D. Taniar and K. Smith (2004). A New Approach of Eliminating Redundant Association Rules. *Database and Expert Systems Applications: 15th International Conference, DEXA 2004*, Zaragoza, Spain, August 30-September 3, 2004. Proceedings. F. Galindo, M. Takizawa and R. Traunmüller. Berlin, Heidelberg, Springer Berlin Heidelberg: 465-474.

Brin, S., R. Motwani, J. D. Ullman and S. Tsur (1997). "Dynamic itemset counting and implication rules for market basket data." *SIGMOD Rec.* 26(2): 255-264.

Hämäläinen, W. (2010). Efficient Discovery of the Top-K Optimal Dependency Rules with Fisher's Exact Test of Significance. *Proceedings of the 2010 IEEE International Conference on Data Mining*, IEEE Computer Society: 196-205.

Hämäläinen, W. (2012). "Kingfisher: an efficient algorithm for searching for both positive and negative dependency rules with statistical significance measures." *Knowledge and Information Systems* 32(2): 383-414.

Leeflang, M., J. J. Deeks, C. Gatsonis and P. M. M. Bossuyt (2008). "Systematic Reviews of Diagnostic Test Accuracy." *Annals of internal medicine* 149(12): 889-897.

Li, J. and O. Zaiane (2015). Associative Classification with Statistically Significant Positive and Negative Rules. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. Melbourne, Australia, ACM: 633-642.

Liu, B., W. Hsu and Y. Ma (2001). Identifying non-actionable association rules. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. San Francisco, California, ACM: 329-334.

McGrane, M. and S. K. Poon (2010). Interaction as an Interestingness Measure. *2010 IEEE International Conference on Data Mining Workshops*.

Piatetsky-Shapiro, G. (1991). Discovery, analysis and presentation of strong rul. *Knowledge Discovery in Databases*. G. a. F. Piatetsky-Shapiro, W. J., AAAI Press: 229--248.

Song, C. and T. Ge (2013). Discovering and managing quantitative association rules. *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. San Francisco, California, USA, ACM: 2429-2434.

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

Verhein, F. and S. Chawla (2007). Using Significant, Positively Associated and Relatively Class Correlated Rules for Associative Classification of Imbalanced Datasets. *Seventh IEEE International Conference on Data Mining* (ICDM 2007).

Webb, G. I. (2006). Discovering significant rules. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. Philadelphia, PA, USA, ACM: 434-443.

Webb, G. I. (2007). "Discovering Significant Patterns." *Machine Learning* 68(1): 1-33.

Webb, G. I. (2010). "Self-sufficient itemsets: An approach to screening potentially interesting associations between items." *ACM Trans. Knowl. Discov. Data* 4(1): 1-20.

Zaki, M. J. (2000). Generating non-redundant association rules. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. Boston, Massachusetts, USA, ACM: 34-43.

Appendices

Number of Rules

The following table describes the average number of rules generated across 10 runs (as described in section 5.1 of the manuscript). Values are reported with their 95% confidence intervals. Alpha value is the goodness threshold used (rules were evaluated using the natural log of Fishers P values).

Dataset	α	No Prune	Classic	Robust Specialisations	Robust (Both)
Aspergillosis	-50	29548.60 ± 4461.94	389.40 ± 31.39	391.60 ± 31.48	268.30 ± 21.10
	-75	4207.80 ± 352.07	119.20 ± 6.36	120.10 ± 6.41	88.90 ± 2.82
	-100	1073.60 ± 124.63	55.90 ± 5.90	56.20 ± 5.87	41.70 ± 3.45
	-125	383.10 ± 33.97	26.90 ± 2.55	27.00 ± 2.57	22.20 ± 1.77
	-150	163.90 ± 16.29	19.50 ± 1.55	19.50 ± 1.55	16.50 ± 1.28
	-175	89.90 ± 6.74	14.60 ± 1.08	14.70 ± 1.04	12.90 ± 1.02
	-200	49.40 ± 5.33	10.00 ± 1.04	10.20 ± 1.10	9.20 ± 0.77
Diabetes	-15	34543.20 ± 12832.45	1327.80 ± 99.71	1676.30 ± 235.23	823.60 ± 83.15
	-20	11816.40 ± 2574.97	613.00 ± 52.78	699.90 ± 73.42	394.40 ± 38.01
	-25	4152.80 ± 227.20	343.40 ± 18.07	365.00 ± 21.57	224.90 ± 10.38
	-30	2731.10 ± 229.94	244.50 ± 10.73	248.60 ± 11.75	162.40 ± 9.20
	-35	1656.20 ± 102.51	180.80 ± 7.58	183.40 ± 7.73	126.30 ± 5.47
	-40	1198.80 ± 86.62	143.60 ± 7.21	145.10 ± 7.38	102.50 ± 4.95
	-45	782.30 ± 78.51	107.10 ± 11.24	107.20 ± 11.29	78.80 ± 7.13
Fertility	-15	362405.60 ± 86789.45	595.30 ± 45.95	618.30 ± 48.06	352.80 ± 25.40
	-20	141740.50 ± 41787.17	278.00 ± 15.74	283.80 ± 15.80	176.80 ± 12.03
	-25	42389.60 ± 11074.22	176.10 ± 5.92	178.20 ± 6.07	111.20 ± 5.71
	-30	20686.90 ± 4609.10	132.50 ± 14.70	133.20 ± 15.04	85.00 ± 9.83
	-35	12736.90 ± 2347.48	113.60 ± 6.68	114.00 ± 7.06	72.80 ± 4.27
	-40	7713.60 ± 1830.05	90.50 ± 7.11	90.80 ± 7.11	62.40 ± 6.12
	-45	4718.80 ± 687.50	74.50 ± 10.29	74.60 ± 10.33	48.70 ± 7.68
Insomnia	-15	12812.70 ± 2399.55	624.00 ± 48.35	747.80 ± 64.69	402.10 ± 33.42
	-20	3180.50 ± 998.68	243.60 ± 37.06	269.00 ± 40.89	161.20 ± 24.88
	-25	876.70 ± 362.33	104.50 ± 15.75	112.50 ± 14.98	72.30 ± 12.27
	-30	271.10 ± 36.97	43.60 ± 6.41	46.80 ± 6.78	32.00 ± 4.42
	-35	136.40 ± 23.64	24.30 ± 5.93	25.40 ± 5.86	17.40 ± 3.59
	-40	59.40 ± 8.13	10.60 ± 1.25	12.40 ± 2.19	8.20 ± 1.20
	-45	31.60 ± 7.02	5.20 ± 1.70	6.10 ± 1.93	4.60 ± 1.4
Mushroom	-1250	61767.80 ± 158.39	409.70 ± 5.32	568.70 ± 7.42	227.40 ± 2.95
	-1375	37501.10 ± 4080.51	308.10 ± 7.70	342.40 ± 10.14	166.00 ± 6.38
	-1500	22634.50 ± 92.39	229.70 ± 5.67	239.40 ± 5.97	125.50 ± 1.67
	-1625	22049.80 ± 39.14	191.30 ± 2.54	196.30 ± 2.85	114.80 ± 1.77
	-1750	19980.00 ± 2498.40	140.70 ± 5.51	141.70 ± 5.51	93.30 ± 6.17
	-1875	7507.80 ± 78.31	88.60 ± 3.75	89.60 ± 3.75	56.70 ± 1.08
	-2000	6430.80 ± 522.10	38.90 ± 5.39	39.90 ± 5.39	34.70 ± 4.15
T10I4D100K	-500	17287.60 ± 191.08	6114.80 ± 53.65	6114.80 ± 53.65	4302.40 ± 43.14
	-750	3484.70 ± 73.63	1568.00 ± 28.03	1568.00 ± 28.03	1217.30 ± 21.15
	-1000	750.40 ± 31.17	411.90 ± 12.19	411.90 ± 12.19	353.50 ± 9.51
	-1250	169.70 ± 3.67	99.80 ± 2.51	99.80 ± 2.51	85.30 ± 2.30
	-1500	76.70 ± 4.39	41.70 ± 2.73	41.70 ± 2.73	36.50 ± 2.47
	-1750	28.50 ± 3.82	16.90 ± 1.55	16.90 ± 1.55	15.60 ± 1.31
	-2000	2.90 ± 1.53	2.90 ± 1.53	2.90 ± 1.53	2.90 ± 1.53
T40I10D100K	-2000	297056.60 ± 16747.78	5477.00 ± 211.44	5477.00 ± 211.44	3675.70 ± 133.93
	-2125	227409.30 ± 30167.75	4165.60 ± 181.86	4165.60 ± 181.86	2874.70 ± 116.69
	-2250	80480.40 ± 31503.32	3001.90 ± 159.61	3001.90 ± 159.61	2195.80 ± 93.08
	-2375	32533.60 ± 24486.95	1746.40 ± 372.79	1746.40 ± 372.79	1323.10 ± 264.55
	-2500	5693.70 ± 611.02	660.20 ± 78.37	660.20 ± 78.37	528.70 ± 69.71
	-2625	1933.20 ± 631.18	341.90 ± 58.24	341.90 ± 58.24	282.20 ± 47.46
	-2750	615.10 ± 272.55	193.50 ± 57.87	193.50 ± 57.87	172.50 ± 51.47

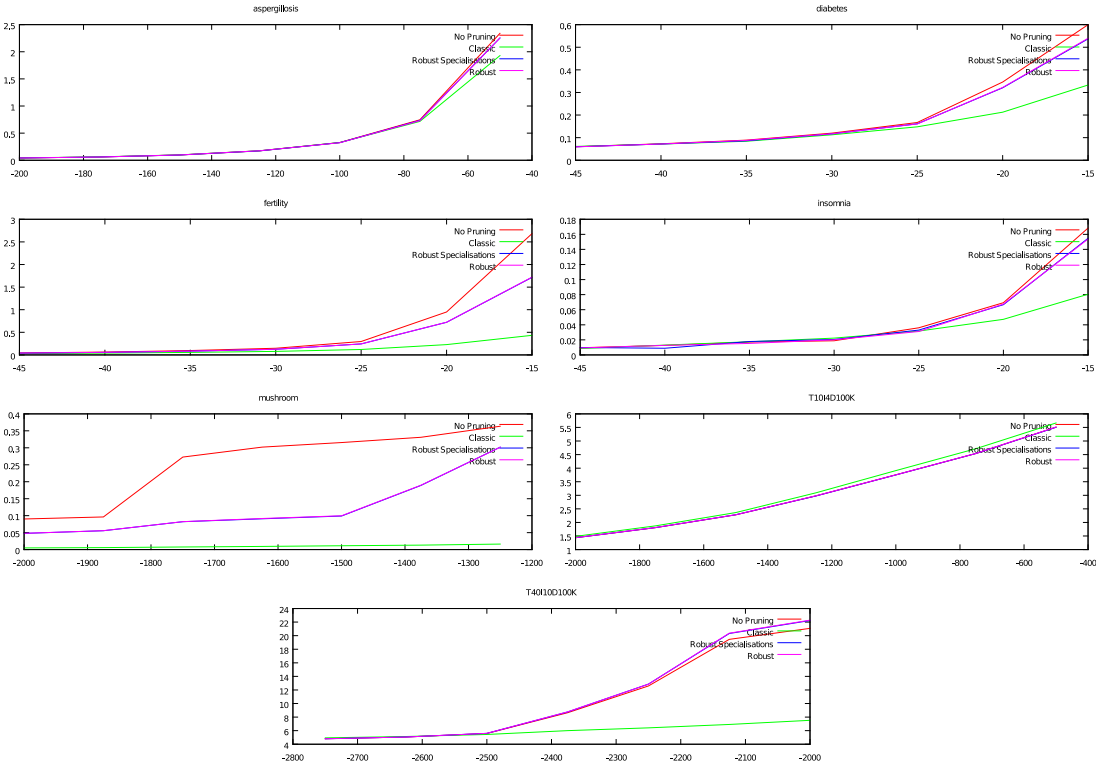
Average P Values

The following table describes the average P value on hold out data for rules on 7 data sets. Results were generated across 10 runs using a 50/50 training/test split (as described in section 5.1 of the manuscript). Values are reported with their 95% confidence intervals. Alpha value is the goodness threshold used (rules were evaluated using the natural log of Fishers P values).

Dataset	α	No Prune	Classic	Robust Specialisations	Robust (Both)
Aspergillosis	-50	-57.44 ± 3.85	-73.19 ± 2.67	-73.25 ± 2.64	-77.29 ± 2.86
	-75	-91.80 ± 3.04	-115.51 ± 2.75	-115.36 ± 2.73	-121.00 ± 2.09
	-100	-125.78 ± 5.52	-155.54 ± 9.36	-155.37 ± 9.22	-165.06 ± 7.93
	-125	-157.39 ± 5.65	-206.80 ± 10.15	-206.60 ± 10.25	-215.07 ± 9.12
	-150	-197.29 ± 8.45	-240.32 ± 7.31	-240.32 ± 7.31	-249.13 ± 7.18
	-175	-224.52 ± 6.51	-265.57 ± 7.95	-265.10 ± 7.75	-273.72 ± 8.71
	-200	-259.87 ± 11.65	-297.70 ± 13.49	-296.33 ± 14.14	-304.60 ± 11.22
Diabetes	-15	-12.17 ± 1.72	-16.78 ± 0.90	-15.30 ± 1.28	-17.67 ± 1.39
	-20	-18.97 ± 2.38	-26.30 ± 1.74	-24.48 ± 1.91	-27.65 ± 2.27
	-25	-30.79 ± 1.34	-36.92 ± 1.58	-35.67 ± 1.53	-39.42 ± 1.54
	-30	-35.59 ± 2.38	-43.34 ± 2.05	-43.11 ± 2.05	-47.00 ± 2.27
	-35	-44.02 ± 1.78	-50.70 ± 1.22	-50.44 ± 1.18	-54.14 ± 1.27
	-40	-48.68 ± 2.62	-55.41 ± 2.06	-55.15 ± 2.07	-58.88 ± 1.94
	-45	-56.41 ± 3.62	-62.17 ± 4.00	-62.14 ± 4.01	-66.32 ± 4.32
Fertility	-15	-14.63 ± 1.86	-20.65 ± 1.35	-20.18 ± 1.29	-21.54 ± 1.23
	-20	-19.21 ± 3.37	-33.41 ± 3.24	-33.00 ± 3.06	-34.13 ± 3.22
	-25	-29.80 ± 3.75	-43.14 ± 3.31	-42.84 ± 3.30	-42.72 ± 3.14
	-30	-34.94 ± 3.51	-54.64 ± 3.28	-54.49 ± 3.30	-55.67 ± 2.92
	-35	-38.45 ± 4.19	-56.66 ± 4.09	-56.55 ± 4.10	-57.15 ± 4.08
	-40	-44.74 ± 5.33	-62.70 ± 6.00	-62.55 ± 5.92	-63.21 ± 5.56
	-45	-50.15 ± 3.88	-66.96 ± 5.25	-66.92 ± 5.28	-66.96 ± 5.36
Insomnia	-15	-10.67 ± 1.41	-13.27 ± 0.74	-12.83 ± 0.80	-13.05 ± 0.72
	-20	-16.50 ± 2.50	-18.77 ± 1.76	-18.68 ± 1.75	-18.61 ± 1.78
	-25	-24.19 ± 3.16	-24.32 ± 1.88	-24.58 ± 1.97	-24.39 ± 2.06
	-30	-31.51 ± 1.70	-30.45 ± 1.52	-30.45 ± 1.64	-30.64 ± 1.62
	-35	-35.02 ± 3.08	-33.93 ± 2.99	-34.37 ± 3.04	-35.13 ± 3.09
	-40	-44.25 ± 4.01	-41.28 ± 2.06	-40.81 ± 2.88	-42.70 ± 2.77
	-45	-57.66 ± 5.25	-55.98 ± 6.25	-55.34 ± 6.45	-58.16 ± 7.72
Mushroom	-1250	-1542.40 ± 8.59	-1604.50 ± 9.79	-1535.21 ± 8.13	-1622.80 ± 9.53
	-1375	-1674.28 ± 39.12	-1700.65 ± 14.68	-1676.63 ± 14.71	-1733.22 ± 20.36
	-1500	-1853.86 ± 17.29	-1793.91 ± 19.56	-1787.94 ± 19.16	-1842.73 ± 17.71
	-1625	-1860.36 ± 11.36	-1843.53 ± 10.95	-1840.13 ± 11.21	-1869.33 ± 10.59
	-1750	-1886.27 ± 24.84	-1903.90 ± 8.54	-1904.92 ± 8.53	-1915.04 ± 9.57
	-1875	-2058.56 ± 15.00	-1996.28 ± 15.97	-1997.05 ± 15.92	-2016.88 ± 13.73
	-2000	-2064.61 ± 13.26	-2040.31 ± 15.01	-2040.44 ± 14.65	-2044.43 ± 13.54
T10I4D100K	-500	-654.97 ± 3.75	-677.78 ± 2.91	-677.78 ± 2.91	-691.13 ± 3.47
	-750	-901.33 ± 7.01	-923.40 ± 5.67	-923.40 ± 5.67	-934.32 ± 6.11
	-1000	-1155.35 ± 11.23	-1171.87 ± 9.15	-1171.87 ± 9.15	-1174.81 ± 8.79
	-1250	-1496.84 ± 15.98	-1489.76 ± 14.07	-1489.76 ± 14.07	-1497.53 ± 14.53
	-1500	-1679.61 ± 30.62	-1690.91 ± 30.69	-1690.91 ± 30.69	-1702.46 ± 30.36
	-1750	-1839.38 ± 44.39	-1865.57 ± 41.17	-1865.57 ± 41.17	-1869.08 ± 40.72
T40I10D100K	-2000	-2205.18 ± 32.30	-2282.00 ± 17.71	-2282.00 ± 17.71	-2302.10 ± 18.75
	-2125	-2201.06 ± 42.97	-2339.99 ± 33.51	-2339.99 ± 33.51	-2360.88 ± 33.30
	-2250	-2305.75 ± 57.95	-2408.56 ± 28.13	-2408.56 ± 28.13	-2423.45 ± 26.10
	-2375	-2410.26 ± 100.15	-2470.00 ± 58.98	-2470.00 ± 58.98	-2481.85 ± 56.55
	-2500	-2585.09 ± 42.48	-2621.87 ± 51.48	-2621.87 ± 51.48	-2631.06 ± 52.50
	-2625	-2691.38 ± 36.23	-2748.60 ± 26.65	-2748.60 ± 26.65	-2756.40 ± 25.11
	-2750	-2686.73 ± 40.39	-2706.56 ± 33.16	-2706.56 ± 33.16	-2710.19 ± 33.49

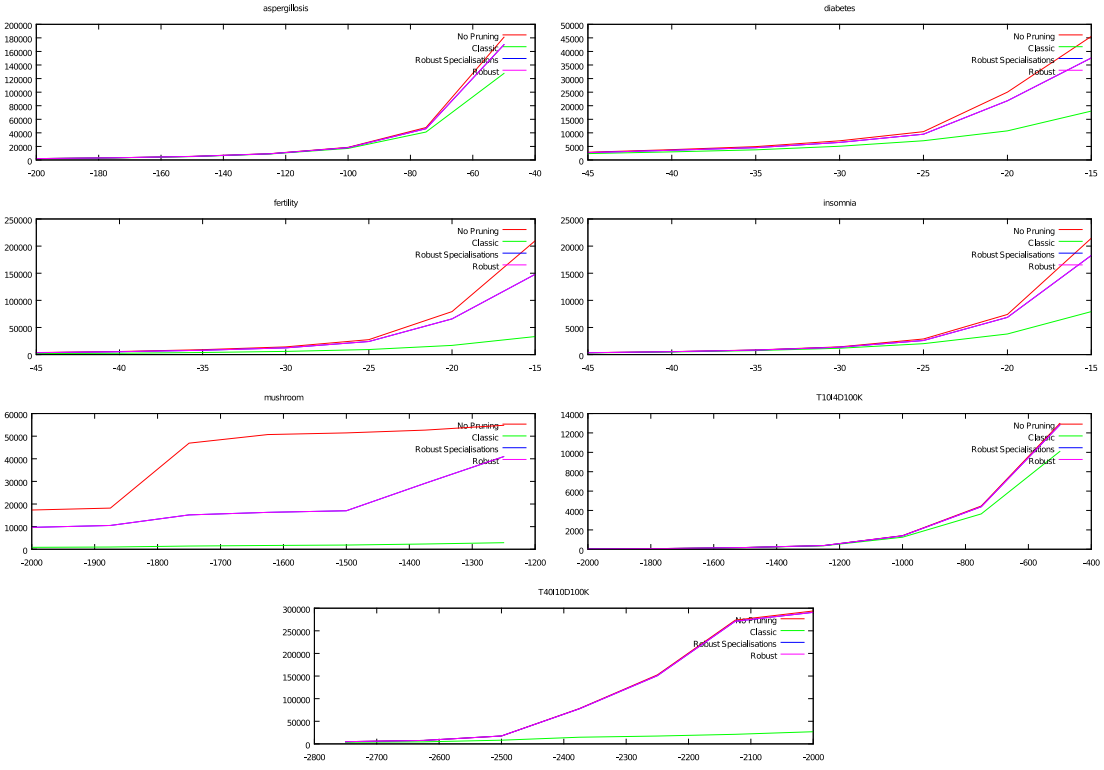
Search Time

Average search time (measured in seconds) vs. goodness threshold (natural log of Fishers P) for 7 data sets.



Number of Search Nodes

Number of search nodes generated vs. goodness threshold (natural log of Fishers P) for 7 data sets.



Peak Memory Usage

Peak memory usage (measured in kb) vs. goodness threshold (natural log of Fishers P) for 7 data sets.