Pre-publication draft of a paper which appeared in the Proceedings of the 2004 Australasian Data Mining Workshop (AusDM04) pp 169-182 Publisher: University of Technology, Sydney

Efficiently Identifying Exploratory Rules' Significance

Shiying Huang and Geoffrey I. Webb

School of Computer Science and Software Engineering
Monash University
Melbourne VIC 3800, Austrailia
{Shiying.Huang, Geoff.Webb}@infotech.monash.edu.au

Abstract. How to efficiently discard potentially uninteresting rules in exploratory rule discovery is one of the important research foci in data mining. Many researchers have presented algorithms to automatically remove potentially uninteresting rules utilizing background knowledge and user-specified constraints. Identifying the significance of exploratory rules using a significance test is desirable for removing rules that may appear interesting by chance, hence providing the users with a more compact set of resulting rules. However, applying statistical tests to identify significant rules requires considerable computation and data access in order to obtain the necessary statistics. The situation gets worse as the size of the database increases. In this paper, we propose two approaches for improving the efficiency of significant exploratory rule discovery. We also evaluate the experimental effect in impact rule discovery which is suitable for discovering exploratory rules in very large, dense databases.

Keyword

Exploratory rule discovery, impact rule, rule significance, interestingness measure

1 Introduction

Exploratory rule discovery techniques seek multiple models which are able to efficiently describe the potentially interesting inter-relationships among attributes in a database. Searching for multiple models instead of a single model often results in numerous spurious or uninteresting rules.

How to automatically discard statistically insignificant rules has been an important issue in research of exploratory rule discovery. Several papers have been devoted to this topic. Bay and Pazzani [4], Liu et. al [10] and Webb [15], developed techniques for identifying insignificant rules with qualitative attributes only (or descretized quantitative attributes). Aumann and Lindell [2] and Huang and Webb [8] both did research on exploratory rule significance with undescretized quantitative attributes as consequent.

When filtering insignificant exploratory rules regarding quantitative attributes, the rule discovery systems have to go through the database several times so as to collect the necessary parameters for the significance test. Moreover, considerable CPU time has to be spent on data access and looking for the set of records which is covered by the antecedent of a rule. For example, it has been shown by Huang and Webb [8] that the time spent for discovering the top 1000 significant impact rules is on the whole much more than that spent on discovering the top 1000 impact rules without using any filter, especially when most of the top 1000 impact rules are insignificant. A technique for improving the efficiency of the insignificance filter is presented in the same paper by introducing the triviality filter. The anti-monotonicity of triviality was utilized to effectively prune the search space.

There is an immediate need for improving the efficiency of the insignificance filter for distributional-consequent exploratory rule discovery, even after the introduction of the triviality filter. In this paper, we propose two approaches for efficiency improving in exploratory rule discovery, which can result in substantial reduction of the computation for discovering significant rules. Although the demonstration is done on impact rule discovery, these techniques can also be recast for other exploratory rule discovery tasks.

The paper is organized as follows: In section 2, we introduce the concept and notations of exploratory rule discovery. Existing techniques for discarding insignificant exploratory rules is introduced in section 3, followed by the brief description of impact rule discovery in section 4. The techniques for improving the efficiency are presented in section 5. In section 6, we provide experimental results and evaluations. Conclusions are drawn in section 7.

2 Exploratory Rule Discovery

Traditional machine learning systems discover a single model from the available data that is expected to maximize the accuracy or some other specific measures of performance on unknown future data. Predictions or classifications are then done on the basis of this single model [15]. Examples include the decision tree [12], the decision rules [11], and the Naive-Bayes classifier. However, alternative models exist that perform equally well as those which are selected by the systems. Thus, it is not always sensible to choose only one of the“best” models in some cases. The criteria for deciding whether a model is best or not also varies with the context of application. Exploratory rule discovery techniques are proposed to overcome this problem by searching for multiple models which satisfy certain constraints and presenting all these models to the user. Thus, the users are provided with alternative choices. Better flexibility is achieved herewith.

Exploratory rule discovery techniques [8] are classified into propositional rule discovery which seeks rules with qualitative attributes or discretized quantitative attributes only and distributional-consequent rule discovery which seeks rules with quantitative attributes as consequent. The status or performance such quantitative attributes are described with their distributions. *Association rule discovery* [1], *contrast sets discovery* [4] are examples of propositional exploratory rule discovery, while *impact rule discovery* [13] and *quantitative association rule* *discovery* [2] both belong to the class of distributional-consequent rule discovery. It is argued that distributional-consequent rules are able to provide better descriptions of the interrelationship between quantitative attributes and qualitative attributes.

Here are some notions of exploratory rule discovery that we are to use in this paper:

1. A *dataset* is a finite set of *records*
2. For propositional rule discovery, a *record* is an element to which we apply Boolean predicates called conditions, while for distributional-consequent rule discovery, a record is a *pair* $< c, v >$, where c is the nonempty set of Boolean conditions, and v is a set of values for the quantitative variables in whose distribution the users are interested.
3. A rule is in the form of $A \to C$. For propositional rules, both A and C are conjunctions of Boolean conditions. The status of such rule is described by interestingness measures like the *support* and the *confidence*. Contrarily, for distributional-consequent rule discovery, A is a conjunction of Boolean conditions while C is a nonempty set of target quantitative variables in which the users are interested. The quantitative variables are described by distributional statistics. We prefer using $A \to target$ to denote a distributional-consequent rule instead, for the purpose of avoiding confusion.
4. Rule $A \to C$ is a parent of $B \to C$ if $A \subset B$. If $|A| = |B| - 1$ than the second rule is a direct parent of the first one, otherwise, it is a grandparent of the first rule.
5. We use the notion $coverset(A)$, where A is a conjunction of conditions, to represent the set of records that satisfy the condition (or set of conditions) A. If a record x is in $coverset(A)$, we say that x is *covered* by A. If A is $\emptyset$, $coverset(A)$ includes all the records in the database.
6. $Coverage(A)$ is the number of records covered by A. $coverage(A) = |coverset(A)|$.

3 Insignificant Exploratory Rules

As is mentioned before, exploratory rule discovery searches for multiple models in a database, and may lead to discovering spurious or uninteresting rules. How to decrease the number of resulting rules becomes a problem of concern. One approach is up to the users to define a suitable set of constraints which may be utilized so that the algorithm can automatically discard some potentially uninteresting rules. Another approach is to perform comparison within resulting rules, so as to present the users with a more compact set of models. Techniques regarding automatically removing potentially uninteresting rules are summarized by Huang and Webb [8].

3.1 Improvement

Filtering insignificant rules using statistical tests is one of the interesting topics of research. By using this technique we perform significance tests among rules and discard those happen to appear interesting only by chance. To provide a clear idea of insignificant rules, we will at first introduce the concept of rule *improvement* defined by Bayardo et al. [5]. *Confidence improvement* which is used as an example, defined a minimum improvement in confidence that a propositional rule must exhibit in order to be regarded as potentially interesting:

$$imp(A \to C) = min(\forall A' \subset A, confidence(A \to C)$$
$$-confidence(A' \to C))$$

It is argued that setting a minimum improvement is desirable in discarding potentially uninteresting exploratory rules. However, the values used for comparison are derived from samples instead of from the total population. There is the problem that the observed improvement provides only an estimate of the true improvement, and if no account is taken of the quality of that estimate, so it is likely to result in poor decisions.

Rule filtering techniques regarding the significance of rules concern about the statistically significance of the improvement, rather than the values of interestingness measures. Statistical tests are done with resulting rules and those within expectation (or without enough surprisingness) are automatically removed. Such techniques may lead to type-1 error, which result in accepting spurious or uninteresting rules and type-2 error, which result in rejecting rules that are not spurious. A technique for statistically sound exploratory rule discovery is proposed by Webb [15] using a holdout set to validate the resulting rules.

3.2 Statistical significance of rules

Chi-square test is a widely used test for identifying propositional rule independence. Liu et al. [10] did research on association rules with a fixed attribute as consequent. They used a chi-square test to decide whether the antecedent of a rule is independent from its consequent or not, accepting only rules whose antecedent and consequent are positively correlated, thus, discarding rules which happen to appear interesting by chance. The rules discarded by using an independent test are referred to as insignificant rules.

Consider the following Boolean-consequent rules:

$$A \to C[support = 60\%, confidence = 90\%]$$
$$A\&B \to C[support = 45\%, confidence = 91\%]$$
$$A\&D \to C[support = 46\%, confidence = 70\%]$$

There is a high possibility that the conditions B and C are conditionally independent given A, thus the second rule provides little interesting information. According to Liu et al., the third rule does not bear interesting information, either. It should also be discarded, because the condition D is negatively correlated to condition C, given A. Bay and Pazzani [4] also made use of Chi-square test to decide the significance of *contrast sets*. Webb [15] proposed a statistically sound technique for filtering insignificant rules, using the Fisher exact test and a hold out set.

Aumann and Lindell [2] and Huang and Webb [8] both proposed ideas for filtering insignificant distributional-consequent exploratory rules. In this paper, we use the definition proposed by the latter.

Definition 1. ***significant impact rule*** *An impact rule $A \rightarrow target$ is significant if the distribution of its target is significantly improved in comparison with the target distribution of any of its direct parents'. The measure for the target distribution can be* the mean, the variance *etc.*

$$significant(A \rightarrow target) = \forall x \in A, dist(coverset(A))$$

$$\gg dist(coverset(A - x) - coverset(A))^1$$

An impact rule is insignificant *if it is not* significant*.*

Definitions of insignificant propositional exploratory rules are provided by Liu et al. [10] and Bay and Pazzani [4].

In this paper, the mean of the target attribute over $coverset(A)$ is used as the interestingness measure to be compared for the impact rule. Statistical test is done to decide whether the target means of two samples are significantly different from each other.

4 K-Most-Interesting Impact Rule Discovery and Notations

The impact rule discovery algorithm we adopt is based on the OPUS [14] algorithm, which enable the successfully discovery of the top k impact rules that satisfy a certain set of constraints.

We characterized the terminology of k-most-interesting impact rule discovery to be used in this paper as follows:

1. An impact rule is in form of $A \rightarrow target$, while the target is describe by the following measures: *coverage*, *mean*, *variance*, *maximum*, *minimum*, *sum* and *impact*.
2. *Impact* is a interestingness measure suggested by Webb [13][2]: $impact(A \rightarrow target) = (mean(A \rightarrow target) - \overline{targ}) \times coverage(A))$.
3. An k-most-interesting impact rule discovery task is a 7-tuple: $KMIIRD(\mathcal{C}, \mathcal{T}, \mathcal{D}, \mathcal{M}, \lambda, \mathcal{I}, k)$.
   - $\mathcal{C}$: is a nonempty set of Boolean conditions, which are the set of available conditions for impact rule antecedents.

[1] The token "$\gg$" is used to denote **significantly improved**, and $dist(\mathcal{R})$ is used to represent the distribution of the target variable over the set of records $\mathcal{R}$.

[2] In this formula, $mean(A \rightarrow target)$ denotes the mean of the *targets* covered by A, and $coverage(A)$ is the number of the records covered by A.

```
Algorithm: OPUS_IR_Filter(Current, Available, 𝓜)

 1. SoFar := ∅
 2. FOR EACH P in Available
    2.1 New := Current ∪ P
    2.2 IF New satisfies all the prunable constraints in 𝓜 except the nontrivial [8]
        constraint THEN
      2.2.1 IF any direct subset of New has the same coverage as New THEN
                  New → relevant stats is a trivial rule
                  Any superset of New is trivial, so do not access any children of this node,
                  go to step 2.
      2.2.2 ELSE IF the mean of New → relevant stats is significantly higher than all its
            direct parents THEN
                  IF the rule satisfies all the other non-prunable constraints in 𝓜
                     THEN record Rule to the ordered rule_list
                  OPUS_IR_Filter(New, SoFar, 𝓜)
                  SoFar := SoFar ∪ P
      2.2.3 END IF
    2.3 END IF
 3. END FOR
```

Table 1. OPUS_IR_Filter

$\mathcal{T}$: is a nonempty set of the variables in whose distribution we are interested.

$\mathcal{D}$: is a nonempty set of records, which is called the database. A record is a pair $< c, v >, c \subseteq C$ and v is a set of values for $\mathcal{T}$.

$\mathcal{M}$: is a set of constraints. There are two types of constraints *prunable* and *unprunable constraints*. *Prunable constraints* are constraints that you can derive useful bounds for search space pruning and still ensures the completeness of information. Examples include the anti-monotone, the succinct constraints [7], or the convertible constraints [9]. Constraints which are not prunable are *unprunable constraints*

λ: $\{X \rightarrow Y\} \times \{D\} \rightarrow \mathcal{R}$ is a function from rules and databases to values and defines a interestingness metric such that the greater the value of $\lambda(X \rightarrow Y, \mathcal{D})$ the greater the interestingness of this rule given the database.

$\mathcal{I}$: is the set of impact rules that can be derived from $\mathcal{D}$, whose antecedents are conjunctions of one or more conditions in C, whose targets are members of $\mathcal{T}$, and which satisfy the constraints in $\mathcal{M}$.

k: is a user specified integer number denoting the number of rules in the ultimate solution for this task.

The original algorithm for impact rule discovery with filters are described in table 1. In this table, *current* is the set of conditions, whose supersets are currently being explored. *Available* is the set of conditions that may be added to *current*. By adding every condition in *available* to *current* one by one, we form the antecedent of the *current rule*: $New \rightarrow target$, which will be referred to later as *current_rules*. *Rule_list* is an ordered list of the top-k interesting rules we have encountered.

5 Efficient Identification of Exploratory Rule Significance

5.1 Deriving Difference Set Statistics without Data Access

According to the algorithm in table 1 and definition 1, we have to compare the mean of current rule with the means of all its direct parents' in order to decide whether a rule is *significant* or not. The set difference operations necessary for this purpose requires excessive data access and computation. However with the status of current rule and all its parent rules known, we will be able to derive the statistics of the difference sets for performing the significance test, without additional access to the database. The following lemma validates this statement.

Lemma 1. *Suppose we are searching for impact rules from a database $\mathcal{D}$. If $A \subset B$, and $coverset(A) - coverset(B) = \mathcal{R}$, where A and B are both conjunction of conditions, $\mathcal{R}$ is a set of records from $\mathcal{D}$. If the mean and variance of the target attribute over $coverset(A)$ and $coverset(B)$ are known, as well as the cardinality of both record sets, the mean and variance of the target attribute over set $\mathcal{R}$ can be derived without additional data access.*

Proof. *Since $coverset(A) - coverset(B) = \mathcal{R}$, it is obvious that*

$$|\mathcal{R}| = coverage(A) - coverage(B) \tag{1}$$

$$mean(\mathcal{R}) = \frac{coverage(A) \times mean(A \rightarrow target) - coverage(B) \times mean(B \rightarrow target)}{|\mathcal{R}|} \tag{2}$$

$$variance(A \rightarrow target) = \frac{\sum_{x \in coverset(A)} (target(x) - mean(A \rightarrow target))^2}{coverage(A) - 1} \tag{3}$$

$$variance(B \rightarrow target) = \frac{\sum_{x \in coverset(B)} (target(x) - mean(B \rightarrow target))^2}{coverage(B) - 1} \tag{4}$$

$$\sum_{x \in coverset(A)} target(x) = mean(A \rightarrow target) \times coverage(A) \tag{5}$$

$$\sum_{x \in coverset(B)} target(x) = mean(B \rightarrow target) \times coverage(B) \tag{6}$$

From 3, 4, 5 and 6 it is feasible to derive the following equation:

$$\begin{aligned} \sum_{x \in \mathcal{R}} target(x)^2 &= \sum_{x \in coverset(A)} target(x)^2 - \sum_{x \in coverset(B)} target(x)^2 \\ &= variance(A \rightarrow target) \times (coverage(A) - 1) \\ &\quad + mean(A \rightarrow target)^2 \times coverage(A) \\ &\quad - variance(B \rightarrow target) \times (coverage(B) - 1) \\ &\quad - mean(B \rightarrow target)^2 \times coverage(B) \end{aligned} \tag{7}$$

$$\sum_{x \in \mathcal{R}} target(x) = \sum_{x \in coverset(A)} target(x) - \sum_{x \in coverset(B)} target(x) \quad (8)$$

Thus,

$$variance(\mathcal{R}) = \frac{\sum_{x\in\mathcal{R}} (target(x) - mean(\mathcal{R}))^2}{|\mathcal{R}| - 1}$$
$$= \frac{\sum_{x\in\mathcal{R}} target(x)^2}{|\mathcal{R}| - 1} - \frac{2mean(\mathcal{R}) \sum_{x\in\mathcal{R}} target(x)}{|\mathcal{R}| - 1} + \frac{|\mathcal{R}| mean(\mathcal{R})^2}{|\mathcal{R}| - 1}$$

Since all the parameters in the right hand side of the equation are already known, we are able to derive all the necessary statistics for doing significance test without accessing the records in $\mathcal{R}$. The lemma is proved.

Note: in this proof, $mean(A \rightarrow target)$ denotes the target mean of the records covered by rule $A \rightarrow target$, $variance(A \rightarrow target)$ denotes the target variance of the records covered by rule $A \rightarrow target$, while $mean(\mathcal{R})$ denotes the target mean of the records in record set $\mathcal{R}$, and $variance(\mathcal{R})$ represents the target variance of the records in $\mathcal{R}$.

By deriving the difference set statistics from the statistics of the *parent_rule* and $New \rightarrow target$ in table 1, we are able to save data access and computation for collecting the statistics for performing the significance test, thus improve the efficiency of the search algorithm.

5.2 The Circular intersection approach

Parallel Intersection Approach According to the definition of significant impact rules, we compare the current rule with all its *direct parents* to identify its significance. In the original OPUS_IR_Filter algorithm, the procedure described in figure 1 is employed to find the *coverset* of every direct parent of the current rule which is being explored. Each arrow in figure 1 represents an intersection operation. When deciding whether a rule with 5 conditions, namely A, B, C, D and E on the antecedent is significant or not, the algorithm have to go through 16 intersection operations! We refer to this approach as the *parallel intersection* approach.

By examining figure 1, we notice that there are considerable overlaps in the *parallel intersection approach*. For example, by using the parallel intersection approach, we have to do the same intersection of $coverset(A)$ and $coverset(B)$ three times, when searching for $coverset(ABCD)$, $coverset(ABCE)$ and $coverset(ABDE)$. There must be a way in which two of these operations can be omitted.

Circular Intersection Approach we propose the approach of *circular intersection* which is shown in figure 2[3]. In this approach, intersections are done in

[3] Each dashed arrow in figure 2 and figure 3 points to the outcome of that specific intersection operation and does not represent an actual operation.

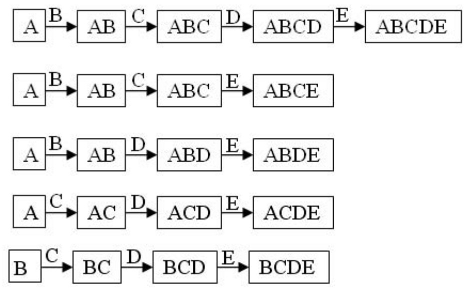

Fig. 1. The parallel intersection Approach for $ABCDE$

two stages. Firstly, in the *forward stage*, intersections are done from condition A to condition E one at a time, and the results are kept in memory. Then we do intersections from the last condition E back to the second one B, which is referred to as the *backward stage*. During the backward stage, the *coverset* of each direct parent of the current rule is found. By introducing the circular intersection approach, the number of intersection operations required for identifying the significance of current rule is reduced to only 10.

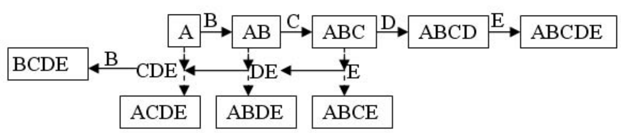

Fig. 2. The circular intersection approach flow for $ABCDE$

Complexity Using the parallel intersection approach, the number of intersection operations for iterating through all the subsets is:

$$(n-2) \times n + 1,$$

where n is the maximum number of conditions on the rule antecedent. The complexity is $O(n^2)$.

After introducing the circular intersection approach, the intersection operations for iterating through all the subsets are:

$$3n - 5.$$

The complexity is $O(n)$. However, practically the difference in running time will not be so dramatic, since we have introduced the triviality filter, which enables the pruning of the search space. Both the parallel intersection procedure and the circular intersection procedure will probably stop at anytime when it is identified that the current rule is a trivial rule.

The two approaches (the difference set statistics derivation approach and the circular intersection approach) mentioned above can combine with each other so as to achieve higher efficiency. We can save one more intersection operation by introducing the difference set statistics derivation technique in section 5.1. Suppose that we are deciding whether the rule $A\&B\&C\&D\&E \rightarrow target$ is significant or not. Now that the statistics of one of its parent $A\&B\&C\&D \rightarrow target$ is known, thus we don't have to derive the statistics of $coverset(ABCD)$ once again. Hereby, one intersection operation can also be saved by following the procedure shown in figure 3 according to lemma 5.1. The number of necessary

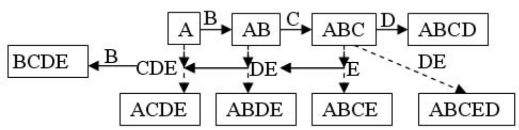

Fig. 3. The circular intersection approach for $ABCDE$ when $current$ is $ABCD$

intersection operations is reduced to

$$3n - 6.$$

The new algorithm for impact rule discovery with filters is shown in table 2. In this table, the *parent_rule* is the corresponding rule for the node whose children we are currently exploring. The antecedent of *parent_rule* is *current*.

6 Experimental Evaluations

In order to explain how the techniques introduced in this paper can practically improve the efficiency of rule discovery, we do our experiments by applying the new algorithm to 10 databases chosen from the UCI Machine Learning repository [6] and the UCI KDD archives [3]. The databases are described in table 3. We applied 3-bin equal-frequency decrepitation to map all the quantitative attributes, except the target attribute, into qualitative ones. The significance level we chose to decide the significance of impact rules is 0.05. The minimum coverage for discovered impact rules is set to 0.01, which is very low. The running time shown in the figures and tables are CPU time spent for the algorithms to search for top 1000 significant impact rules with the highest impact on a computer with two PIII 933MHz processors, 1.5G memory, and 4G virtual memory.

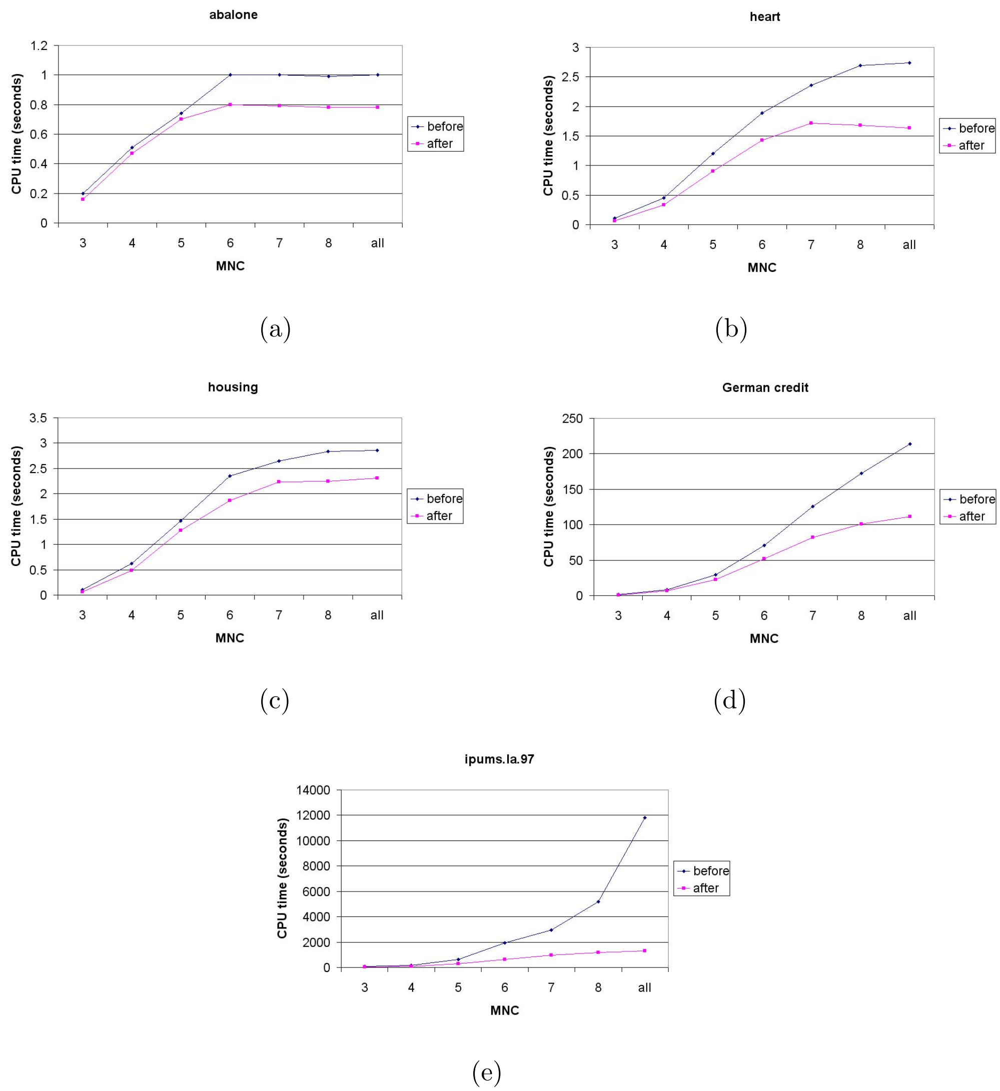

Fig. 4. Comparison of Running Time before and after applying data access saving techniques for (a) *abalone*, (b) *heart*, (c) *housing*, (d) *German credit*, and (e) *ipums.la.97*

```
Algorithm: OPUS_IR_Filter(Current, Available, parent_rule, 𝓜)

  1 SoFar := ∅;
  2 FOR EACH P in Available
    2.1 New := Current ∪ P
    2.2 IF New satisfies all the prunable constraints in 𝓜 except the nontrivial
        constraint THEN
      2.2.1 Derive the statistics of coverset(Current) − coverset(New), according to lemma
            5.1.
      2.2.2 IF the mean of New → target is not significantly improved comparing to
            coverset(Current) − coverset(New) THEN
                go to step 2.2.4;
      2.2.3 ELSE use the circular intersection to comparing the mean of New → target with
            the mean of its direct parents other than parent_rule
        2.2.3.1 IF the mean New → target is significantly improved comparing to all its
                direct parents THEN
                record New → target to rule_list;
                OPUS_IR_Filter(New, SoFar, New → target);
                SoFar := SoFar ∪ P ;
        2.2.3.2 END IF;
      2.2.4 END IF;
    2.3 END IF;
  3 END FOR
```

Table 2. Improved OPUS_IR_Filter

database	records	attributes	conditions	Target
Abalone	4117	9	24	Shuckedweight
Heart	270	13	40	Max heart rate
Housing	506	14	49	MEDV
German credit	1000	20	77	Credit amount
Ipums.la.97	70187	61	1693	Total income
Ipums.la.98	74954	61	1610	Total income
Ipums.la.99	88443	61	1889	Total income
Ticdata2000	5822	86	771	Ave. income
Census income	199523	42	522	Wage per hour
Covtype	581012	55	131	Elevation

Table 3. Basic information of the databases

We ran the program without using the algorithm proposed in table 1 first. For databases *abalone, heart, housing, German credit* and *ipmus.la.97*, which is relatively smaller, we set the maximum number of conditions on the rule antecedent (MNC for short) from 3 to 8, and then run the program with no limit on the MNC. After that, the new algorithm in table 2 is ran according to the same procedure. The CPU time spent for these programs to search for the top 1000 significant impact rules is presented using line charts in figure 4. For *ipmus.la.98, ipmus.la.99, ticdata2000, census income* and *covtype*, which are relatively larger databases, we only ran the programs with MNC set to 3, 4, and 5. The experimental results are listed in table 4.

With *MNC* set to 3, the number of intersection operations required for doing insignificant tests are the same, regardless of whether the circular intersection technique is introduced or not. Thus, the difference in efficiency between the

Database	status	MNC=3	MNC=4	MNC=5
Ipums.la.98	before	74.41	300.47	1860.31
	after	46.15	130.62	482.52
Ipums.la.99	before	750.6	2785.46	9805.81
	after	103.29	312.66	820.72
Ticdata2000	before	116.55	1669.76	10808.03
	after	73.17	1027.33	7946.36
Census-income	before	577.32	2362.53	3781.6
	after	351.56	1054.58	2075.2
Covtype	before	3529.95	11300.45	20686.95
	after	2315.47	9803.97	16987.18

Table 4. Time spent (in seconds) for searching for significant rules in databases: *ipums.la.98, ipums.la.99, ticdata2000, census income, covtype* before and after the techniques are introduced

algorithms in table 1 and table 2 is caused by applying the data access saving approach which is proposed in section 5.1. For instance, it took the algorithm in table 1 more than 70 seconds to find the top 1000 significant rules in *ipums.la.98* with MNC set to 3, while the time for the algorithm in table 2 to finish the same task is only 57 seconds.

When the MNC is set to a number greater than 3, the trend of increase in running time is much steeper before applying the techniques proposed in section 5 than after. The difference in efficiency increases with the MNC. When there is no limit on the maximum number of conditions on the rule antecedent, the time spent for the new algorithm to search for top 1000 significant impact rules in *ipums.la.97* is less than one sixth of that necessary for the old one. However, the running time is also influenced by other factors including the size of the databases, the number of trivial rules in the top 1000 impact rule, and the number of significant rules.

7 Conclusion

The large number of resulting rules has long been a handicap for exploratory rule discovery. Many techniques have been proposed to reduce the set of resulting rules to a manageable size. Removing statistically insignificant rules is one of those techniques that are popular. Such techniques lead to considerable decrease in the resulting number of exploratory rules. However, performing statistical tests to identify the significance of a rule requires considerable data access and computation. We proposed two techniques in this paper, which can improve the efficiency of rule discovery by deriving difference set statistics without additional reference to the data, and by reducing the redundancy of intersection operations. We implemented the techniques in k-most-interesting impact rule discovery, which is suitable for distributional-consequent exploratory rule discovery in very large, dense databases. Experimental results show a substantial improvement in efficiency after applying these techniques.

References

1. Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 26–28 1993.
2. Y. Aumann and Y. Lindell. A statistical theory for quantitative association rules. In *Knowledge Discovery and Data Mining*, pages 261–270, 1999.
3. S. D. Bay. The uci kdd archive [http://kdd.ics.uci.edu], 1999.
4. S.D. Bay and M.J. Pazzani. Detecting group differences: Mining contrast sets. In *Data Mining and Knowledge Discovery*, pages 213–246, 2001.
5. Roberto J. Bayardo, Jr., Rakesh Agrawal, and Dimitrios Gunopulos. Constraint-based rule mining in large, dense databases. *Data Min. Knowl. Discov.*, 4(2-3):217–240, 2000.
6. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
7. J. Han and M. Kamber. *Data mining : concepts and techniques*. Morgan Kaufmann, 2001.
8. Shiying Huang and Geoffrey I. Webb. Discarding insignificant rules during impact rule discovery in large database, 2004.
9. Jiawei han Jian Pei and Laks V.S. Lakshmanan. Mining frequent itemsets with convertible constraints. In *Proceedings of the 17th International Conference on Data Engineering*, page 433. IEEE Computer Society, 2001.
10. B. Liu, W. Hsu, and Y. Ma. Pruning and summarizing the discovered associations. In *Knowledge Discovery and Data Mining*, pages 125–134, 1999.
11. R. S. Michalski. A theory and methodology of inductive learning. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 83–134. Springer, Berlin, Heidelberg, 1984.
12. J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.
13. G. I. Webb. Discovering associations with numeric variables. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 383–388. ACM Press, 2001.
14. Geoffrey I. Webb. OPUS: An efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research*, 3:431–465, 1995.
15. G.I. Webb. Statistically sound exploratory rule discovery, 2004.