On the Discovery of Significant Statistical Quantitative Rules

Hong Zhang
OPIM Department
The Wharton School
University of Pennsylvania
Philadelphia, PA 19104
hongz@wharton.upenn.edu

Balaji Padmanabhan
OPIM Department
The Wharton School
University of Pennsylvania
Philadelphia, PA 19104
balaji@wharton.upenn.edu

Alexander Tuzhilin
IOMS Department
Leonard N. Stern School of Business
New York University
New York, NY 10012
atuzhili@stern.nyu.edu

ABSTRACT
In this paper we study market share rules, rules that have a certain market share statistic associated with them. Such rules are particularly relevant for decision making from a business perspective. Motivated by market share rules, in this paper we consider statistical quantitative rules (SQ rules) that are quantitative rules in which the RHS can be any statistic that is computed for the segment satisfying the LHS of the rule. Building on prior work, we present a statistical approach for learning all significant SQ rules, i.e., SQ rules for which a desired statistic lies outside a confidence interval computed for this rule. In particular we show how resampling techniques can be effectively used to learn significant rules. Since our method considers the significance of a large number of rules in parallel, it is susceptible to learning a certain number of "false" rules. To address this, we present a technique that can determine the number of significant SQ rules that can be expected by chance alone, and suggest that this number can be used to determine a "false discovery rate" for the learning procedure. We apply our methods to online consumer purchase data and report the results.

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Database Applications - Data Mining.

General Terms
Algorithms, Management.

Keywords
Rule discovery, market share rules, statistical quantitative rules, nonparametric methods, resampling.

1. INTRODUCTION
Rule discovery is widely used in data mining for learning interesting patterns. Some of the early approaches for rule learning were in the machine learning literature [11, 12, 21]. More recently there have been many algorithms [1, 25, 28, 31] proposed in the data mining literature, most of which are based on the concept of association rules [1]. While all these various approaches have been successfully used in many applications [8, 22, 24], there are still situations that these types of rules do not capture. The problem studied in this paper is motivated by market share rules, a specific type of rule that cannot be represented as association rules. Informally, a market share rule is a rule that specifies the market share of a product or a firm under some conditions.

The results we report in this paper are from real user-level Web browsing data provided to us by comScore Networks. The data consists of browsing behavior of 100,000 users over 6 months. In addition to customer specific attributes, two attributes in a transaction that are used to compute the market share are the site at which a purchase was made and the purchase amount. Consider the example rules below that we discovered from the data:

(1) Household Size = 3 $\wedge$ 35K < Income < 50K $\wedge$ ISP = Dialup $\Rightarrow$ $\text{marketshare}_{\text{Expedia}}$ = 27.76%, support = 2.1%

(2) Region = North East $\wedge$ Household Size = 1 $\Rightarrow$ $\text{marketshare}_{\text{Expedia}}$ = 25.15%, support = 2.2%

(3) Education = College $\wedge$ Region = West $\wedge$ 50 < Household Eldest Age < 55 $\Rightarrow$ $\text{marketshare}_{\text{Expedia}}$ = 2.92%, support=2.2%

(4) 18 < Household Eldest Age < 20 $\Rightarrow$ $\text{marketshare}_{\text{Expedia}}$ = 8.16%, support = 2.4%

The market share for a specific site, e.g. Expedia.com, is computed as the dollar value of flight ticket purchases (satisfying the LHS of the rule) made at Expedia.com, divided by the total dollar value of all flight ticket purchases satisfying the LHS. The discovered rules suggest that Expedia seems to be doing particularly well among the single households in the North East region (rule 2), while it cedes some market in the segment of teenagers (rule 4). Rules such as these are particularly relevant for business since they suggest natural actions that may be taken. For example, it may be worth investigating the higher market share segments to study if there is something particularly good that is being done, which is not being done in the lower market share segments.

More generally, "market share" is an example of a statistic that is computed based on the segment satisfying the antecedent of the rule. Besides market share, various other quantitative statistics on the set of transactions satisfying the LHS of a rule can be computed, including mean and variance of an attribute. Prior

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

work on learning quantitative association rules [2, 33] studied the discovery of rules with statistics such as the mean, variance, or minimum/maximum of a single attribute on the RHS of a rule. In this paper we generalize the structure of the rules considered in [2] to rules in which the RHS can be any quantitative statistic that can be computed for the subset of data satisfying the LHS. This statistic can even be computed based on multiple attributes. We term such rules as *statistical quantitative rules* (*SQ rules*).

With respect to learning SQ rules from data, we formulate the problem as learning *significant* SQ rules that have adequate support. We define an SQ rule to be significant if the specific statistic computed for the rule lies outside a certain confidence interval. This confidence interval represents a range in which the statistic can be expected by *chance alone*. This is an important range to identify if the rules discovered are to be interpreted as suggesting fundamental relationships between the LHS and the market share. For example, by chance alone if it is highly likely that the market share of Expedia is between 25% and 30% for any subset of data, then it is not at all clear that the rule relating income and Expedia's market share (rule 1 in the example) is identifying a fundamental relationship between income and the market share.

While prior work [6, 9] has used confidence intervals to identify significant rules, most of these approaches are either parametric or specific for binary data. Building on prior work in this paper we present a statistical approach for learning significant SQ rules that is entirely non-parametric. In particular we show how resampling techniques, such as permutation, can be effectively used to learn confidence intervals for rules. Based on these confidence intervals, significant rules can be identified. However, since our method considers the significance of a large number of rules in parallel, for a given significance level it is susceptible to learning a certain number of *false* rules. To address this we present an intuitive resampling technique that can determine the number of false rules, and argue that this number can be used to determine a "false discovery rate" for the learning procedure. The practical significance of this approach is that we learn significant SQ rules from data and specify what the false discovery rate exactly is.

The paper is organized as follows. We first define SQ rules in the next section. Section 3 presents an algorithm for computing confidence intervals and Section 4 presents an algorithm for learning significant SQ rules. In Section 5 we explain how the false discovery rate for our approach can be computed. We present detailed experimental results on real web browsing data in Section 6 followed by a literature review and conclusions.

2. STATISTICAL QUANTITATIVE RULES

In this section we define SQ rules and significant SQ rules. Let $A = \{A_1, A_2, \ldots, A_n\}$ be a set of attributes that will be used to describe segments and $B = \{B_1, B_2, \ldots, B_m\}$ be another set of attributes that will be used to compute various statistics that describe the segment. Let $dom(A_i)$ and $dom(B_j)$ represent the set of values that can be taken by attribute A_i and B_j respectively, for any $A_i \in A$ and $B_j \in B$. Let D be a dataset of N transactions where each transaction is of the form $\{A_1 = a_1, A_2 = a_2, \ldots, A_n = a_n, B_1 = b_1, B_2 = b_2, \ldots, B_m = b_m\}$ where $a_i \in dom(A_i)$ and $b_j \in dom(B_j)$. Let an atomic condition be a proposition of the form $value_1 \leq A_i \leq value_2$ for ordered attributes and $A_i = value$ for unordered attributes where $value$, $value_1$, $value_2$ belong to the finite set of discrete values taken by A_i in D. Finally, let an itemset represent a conjunction of atomic conditions.

Definition 2.1 (SQ rule). Given (i) sets of attributes A and B, (ii) a dataset D and (iii) a function f that computes a desired statistic of interest on any subset of data, an SQ rule is a rule of the form:

$$X \Rightarrow f(D_X) = statistic, support = sup^{1} \quad (2.1)$$

where X is an itemset involving attributes in A only, D_X is the subset of D satisfying X, the function f computes some statistic from the values of the B attributes in the subset D_X, and support is the number of transactions in D satisfying X. □

Note that the statistic on the RHS of the rule can be computed using the values of multiple attributes. The following examples are listed to demonstrate different types of rules that an SQ rule can represent. For ease of exposition we use the name of the desired statistic in the RHS instead of referring to it as $f(D_X)$.

1. ***Quantitative association rules*** [2]:

population-subset ⇒ mean or variance values for the subset (2.2)

Quantitative association rules are a popular representation for rules in the data mining literature in which the RHS of a rule represents the mean or variance of some attribute. Example: *Education = graduate* ⇒ *Mean*(*purchase*) = $15.00. (2.2) is a special case of (2.1), where *f*(*subset*) is the mean of some attribute B_j in the subset of data.

2. *Market share rules*:

Let $\{A_1, A_2, \ldots, A_n, MSV, P\}$ be a set of attributes in a dataset D. *MSV* (Market Share Variable) is a special categorical attribute for which the market share values are to be computed. P is a special continuous variable that is the basis for the market share computation for *MSV*. For example, each transaction T_k may represent a book[2] purchased online. A_1 through A_n may represent attributes of the customer who makes the purchase, such as income, region of residence and household size. For each transaction, *MSV* is the variable indicating the online book retailer where the purchase was made. *dom(MSV)* may be {Amazon, Barnes&Noble, Ebay} and P is the price of the book purchased. For a specific $v \in dom(MSV)$ a market share statistic can be computed as described below. Market share rules have the following form:

$$X \Rightarrow marketshare_v = msh, support = sup \quad (2.3)$$

where X is an itemset consisting of attributes in $\{A_1, A_2, \ldots, A_n\}$ and $marketshare_v$ is a statistic that represents the market share of a specific $v \in dom(MSV)$. This is computed as follows. Let D_X represent the subset of transactions satisfying X and $D_{X,\ MSV=v}$

[1] In association rules, *support* is the number of transactions satisfying **both** LHS and RHS of a rule. In SQ rules, since the RHS is not an itemset, we define *support* as the number of transactions satisfying the LHS of a rule **only**.

[2] The provider, comScore Networks categorizes each purchase into categories such as "book", "travel" and "consumer electronics". Hence we can generate datasets in which all transactions represent purchases in a single category, and this helps in the generation of market share rules representing specific categories.

represent the subset of transactions satisfying ($X \wedge MSV = v$). Then $marketshare_v$ is computed as $sum(P, D_{X, MSV=v}) / sum(P, D_X)$, where $sum(P, D)$ is the sum of all the values of attribute P in the transactions in D.

Market share rules naturally occur in various applications, including online purchases at various Web sites, sales applications, and knowledge management applications. The examples presented in the introduction are real market share rules discovered in online purchase data. The following additional examples illustrate the versatility and usefulness of market share rules.

- Within a specific product category (e.g. shoes) Footlocker sells competing brands of shoes. In their transaction data, the brand of the shoe can be the *MSV* and the purchase price is *P*.
- Consider a dataset of patents associated with some area (e.g. hard disks). Each record may consist of several attributes describing a patent, including one attribute (*MSV*) which represents the organization to which the patent belongs and another attribute that is always 1 (representing *P* and indicating a granted patent) in the data. For a specific organization, e.g. IBM, market share rules will represent the percentage of patents that belong to IBM under some conditions involving other attributes of the patent.

Definition 2.1 differs from the definition of quantitative rule [2, 33] as follows. First, it is not limited to mean and variance statistics and assumes a much broader class of statistics, including the market share statistics. Second, unlike quantitative rules, the statistic of interest in the RHS of a rule can be computed based on *multiple* attributes.

Definition 2.2 (Significant SQ rule). For a given significance level $\alpha \in (0, 1)$, let $(stat_L, stat_H)$ be the $(1 - \alpha)$ confidence interval for a desired statistic, where this confidence interval represents the range in which the statistic can be expected by chance alone. An SQ rule $X \Rightarrow f(D_X) = statistic$, $support = sup$ is significant if *statistic* lies outside the range $(stat_L, stat_H)$. □

The main objective of this paper is to discover all significant SQ rules. The first challenge in learning significant SQ rules is in constructing a confidence interval for the desired statistic such that this interval represents a range of values for the RHS statistic that can be expected by chance alone. In the next section we present an algorithm for learning these confidence intervals.

3. COMPUTING CONF. INTERVALS

The first question that needs to be addressed is what is meant by "a range for the statistic that can be expected by chance alone". In this section we start by addressing this question and outline a procedure by which such a range can be computed. Next we will point out the computational challenge in implementing such a procedure for learning these intervals for several SQ rules and then outline three observations that will substantially help address the computational problems. Based on these observations we present a resampling-based algorithm for computing the confidence intervals.

3.1 Motivation and outline of a procedure

For a given SQ rule, the desired confidence interval theoretically represents the range in which the statistic can be expected when there is *no* fundamental relationship between the LHS of the rule and the statistic. More precisely, since the statistic is computed from the values of the *B* attributes, the confidence interval represents the range in which the statistic can be expected when the *A* attributes are truly independent of the *B* attributes.

Without making any parametric distributional assumptions, such a confidence interval can be generated using the classical non-parametric technique of permutation. Indeed permutation-based approaches have been commonly used to generate confidence intervals in the statistics literature [16]. If R is the set of all attributes in a dataset, the basic idea in permutation is to create multiple datasets by randomly permuting the values of some attributes $R_i \subset R$. Such a permutation would create a dataset in which R_i is independent of $(R - R_i)$, but would maintain the distributions of R_i and $(R - R_i)$ in the permutation dataset to be the same as the distributions of these attributes in the original dataset.

Table 3.1 illustrates one example of a permutation dataset D' in which the *B* attributes are randomly permuted. Since a desired statistic can be computed on each permutation dataset, a distribution for the statistic can be computed based on its values from the multiple permutation datasets. A confidence interval can then be computed from this distribution.

Table 3.1 Dataset permutation

Original dataset D: → Permutation dataset D':

A_1	A_2	B_1	B_2
1	2	***3***	***8***
1	3	***5***	***6***
2	3	***7***	***4***

→

A_1	A_2	B_1	B_2
1	2	***5***	***6***
1	3	***7***	***4***
2	3	***3***	***8***

As mentioned above, this is a commonly used procedure in non-parametric statistics. The reason this procedure makes sense is as follows. Even if there is a relationship between the LHS of an SQ rule and the *statistic* on the RHS, by holding the *A* attributes fixed and randomly re-ordering the values of the *B* attributes the relationship is destroyed and the *A* attributes and *B* attributes are now independent of each other. Repeating this procedure many times provides many datasets in which the *A* attributes and *B* attributes are independent of each other, while maintaining the distributions of the *A* and *B* attributes to be the same as their distributions in the original dataset. The values for the statistic computed from the many permutation datasets is used to construct a distribution for the statistic that can be expected when the *A* attributes are truly independent of the *B* attributes.

Specifically, for the same itemset X, compare the following two SQ rules in D and D',

$$D: X \Rightarrow f(D_X) = stat_D, support = sup_D \quad (3.1)$$

$$D': X \Rightarrow f(D'_X) = stat_{D'}, support = sup_{D'} \quad (3.2)$$

First note that the supports of the rules are the same since the number of records satisfying X in the permutation dataset is the same as the original dataset. We will use this observation to build a more efficient method for computing confidence intervals

shortly. A confidence interval for the rule in (3.1) can be computed using the following naïve procedure.

1. Create permutation dataset D' from the original dataset D and compute $stat_{D'}$ (as mentioned earlier in Section 2, the function f computes this number based on the records satisfying X).
2. Repeat step 1 $N_{perm} > 1000$ times[3], sort all the N_{perm} $stat_{D'}$ values in an ascending order ($stat_{D'\text{-}1}$, $stat_{D'\text{-}2}$,…, $stat_{D'\text{-}Nperm}$) and let the $\alpha/2^{th}$ and $(1 - \alpha/2)^{th}$ percentiles[4] from this list be $stat_{D'\text{-}L}$ and $stat_{D'\text{-}H}$. The N_{perm} values computed above represents a distribution for the statistic that can be expected by chance alone, while the percentile values from this distribution determine a specific confidence interval. (Below we use the terms "distribution" and "confidence interval" frequently.)
3. The $(1 - \alpha)$ confidence interval for the SQ rule in Equation (3.1) is $(stat_{D'\text{-}L}, stat_{D'\text{-}H})$.

3.2 Computational challenges and solutions

Computing these confidence intervals for multiple candidate SQ rules creates several computational problems which we will address in this section. For example, if we need to test 10,000 potential significant rules (which is a reasonably small number for data mining tasks), then we would need to repeat the above steps 10,000 times, and this means generating permutation datasets $10{,}000 \times N_{perm} > 10^7$ times and to compute the desired statistic in each permutation dataset.

The following observations substantially reduce the computational complexity of the procedure.

1. *Sampling can be used instead of creating permutation datasets*. For the SQ rule in Equation (3.1), computing $stat_{D'}$ on a permutation dataset is really equivalent to computing $stat_{D'}$ based on a random sample of sup_D records in D. This is the case since none of the A attributes play a role in the computation of the statistic. Permuting the dataset, identifying the (sup_D) records where X holds, and then computing the statistic on this subset achieves the same effect as picking a random sample of sup_D records from D and computing the statistic on this random subset. Hence to determine the confidence interval for the SQ rule in Equation (3.1), instead of permuting the dataset N_{perm} times, it is enough to sample sup_D records from D for N_{perm} times.

2. *Some of the candidate SQ rules have the same support values as other rules*. Based on this observation, confidence intervals for two SQ rules with the same support can be approximated by the same interval. This is the case since for a given rule the interval is generated by sampling sup_D records many times and if another rule has *support* = sup_D then the interval for that rule will be similar if the same procedure is repeated (it will not be exactly the same because of randomization). Therefore, fewer confidence intervals need to be generated.

3. *It is adequate to generate a fixed number of intervals, independent of the number of rules considered*. We observe that the interval for an SQ rule with *support* = sup_D can be approximated by an interval computed by sampling sup_E records where sup_E is "reasonably close" to sup_D. This is a heuristic that we use to considerably reduce the complexity of the procedure. Denote N_{Rule} as the number of rules to be tested. If all rules have different *support* values, we need to construct N_{Rule} distributions. Instead, we would construct a fixed number N_{dist} distributions, such that for rule "$X \Rightarrow f(D_X)$ = *statistic*, *support* = *sup*", *statistic* is compared with the distribution that is constructed by sampling the closest number of transactions to *sup*. This heuristic is more meaningful when we consider *support* in terms of percentage of transactions satisfying LHS of a rule, which is a number between 0 and 1.

3.3 Algorithm *CIComp*

Based on the above observations, we present in Figure 3.1 algorithm *CIComp* for constructing N_{dist} distributions and determining the $(1 - \alpha)$ confidence intervals for a given significance level.

Input: dataset D with N transactions, the number of distributions N_{dist}, the number of points in each distribution N_{perm}, a function f that computes the desired statistic, and significance level α.

Output: N_{dist} distributions and significance thresholds.

```
1   for ( dist = 1; dist ≤ N_dist; dist++ ) {
2       N_sample = dist/N_dist × N;
3       for ( perm = 1; perm < N_perm; perm++ ) {
4           S = N_sample transactions from D sampled without
                  replacements[5];
5           stat[dist][perm] = f(S);
6       }
7       sort(stat[dist]);
8       LowerCI[dist] = stat[dist][(N_perm + 1) × α/2];
9       UpperCI[dist] = stat[dist][(N_perm + 1) × (1 – α/2)];
10  }
11  Output stat[][], LowerCI[], UpperCI[]
```

Figure 3.1 Algorithm *CIComp*

In the above algorithm, N_{dist}, N_{perm}, and α are user-defined parameters. α is usually chosen to be 5%, 2.5% or 1%. For N_{dist} and N_{perm}, the larger they are, the more precise the distributions will be. Let N = 1000, N_{dist} = 100, N_{perm} = 999, and α = 5%. We use these numbers as an example to explain the algorithm. For step 2, the first distribution corresponds to $N_{sample} = dist/N_{dist} \times N$ = 1/100 × 1000 = 10 transactions. Step 3 to 6 computes N_{perm} = 999 statistics for 10 randomly sampled transactions from dataset D. Then we sort these 999 statistics and pick $\alpha/2$ and $1 - \alpha/2$ percentiles, which are the 25th and 975th numbers in the distribution, as the lower and upper thresholds for the $(1 - \alpha)$ confidence interval. Steps 2 through 9 are repeated N_{dist} = 100 times to get the desired number of distributions and confidence intervals.

[3] N_{perm} is typically a big number. If we let $N_{perm} = N!$, which is the number of all possible permutations, we will be implementing a Monte Carlo test. On large datasets, such a test is impractical. For a *statistic* like market share whose value is limited by 0 and 1, $N_{perm} > 1000$ makes the distribution precise to the third decimal place. In our experiments, N_{perm} = 1999.

[4] Since we do not have any prior assumption about the expected value of the statistic we use a two sided *p*-value.

[5] If the sampling is done with replacement then the interval will be the bootstrap confidence interval. The two intervals will essentially be the same when the support of the itemset is small.

The computation complexity of the algorithm in Figure 3.1 is O($N \times N_{perm} \times N_{dist}$), whereas the complexity of naïve method is O($N \times N_{perm} \times N_{rule}$). Note that N_{dist} can be fixed to a reasonable small number, e.g. 100, whereas N_{rule} is the number of rules that are being tested and can easily be orders of magnitude more than N_{dist}.

4. DISCOVERING SQ RULES

Given the distributions and confidence intervals, discovering all significant statistical rules is straightforward. Algorithm *SigSQrules* is presented in Figure 4.1.

Input: dataset D with N transactions, sets of attributes A and B, N_{dist}, $stat$[][], $LowerCI$[], and $UpperCI$[] from algorithm *CIComp*, a function f that computes the desired statistic, minimum support *minsup* and a large itemset generation procedure *largeitemsets*.

Output: set of α Significant rules, *sigrules*.

```
1   L = largeitemsets(D, A, minsup) # generates large
    itemsets involving attributes in A
2   sigrules = {}
3   forall (itemsets x ∈ L) {
4       x.stat = f(D_x) // statistic computed on
              transactions satisfying x
5       dist = round(support(x) / N × N_dist)
6       if x.stat ∉ (LowerCI[dist], UpperCI[dist]) {
              // x ⇒ f(D_x) = x.stat is significant
7             x.pvalue = 2 × percentile of x.stat in
                      stat[dist][1..N_perm]
8             sigrules = sigrules ∪ { x ⇒ f(D_x) = x.stat,
                      support = support(x)}
9       }
10  }
```

Figure 4.1 Algorithm *SigSQrules*

Given N_{dist} distributions constructed from the algorithm *CIComp*, we use the above algorithm to discover all significant SQ rules. We continue to use the example N = 1000, N_{dist} = 100, and N_{perm} = 999 to describe the steps in Figure 4.1. Note that the attributes in A represent the attributes in the dataset that are used to describe segments for which statistics can be computed. Step 1 uses any large itemset generation procedure in rule discovery literature to generate all large itemsets involving attributes in A. The exact procedure used will depend on whether the attributes in A are all categorical or not. If they are, then *Apriori* algorithm can be used to learn all large itemsets. If some of them are continuous then other methods such as the ones described in [31] can be used.

Step 4 computes the *statistic* function for each large itemset, x. In step 5, we find out which distribution is to be used for significance test. For example, if *support*(x) = 23, then *support*(x)/$N \times N_{dist}$ = (23/1000) × 100 = 2.3, and hence *dist* will be *round*(2.3) = 2. We would compare *x.stat* with its corresponding confidence interval (*LowerCI*[2], *UpperCI*[2]) in step 6. If *x.stat* is outside of the confidence interval, the rule is significant, and we use step 7 to calculate its 2-side p-value. If *x.stat* is the qth percentile, the 2-side p-value is 2 × min(q%, 1–q%). The p-value is not only a value to understand how significant a rule is, but is also useful for determining the false discovery rate in Section 5. Note that the confidence interval used to test significance of a rule is approximate since we do not compute this interval for the exact value of the support of this rule. Instead we use the closest interval (which was pre-computed as described in Section 3.2) corresponding to this support value. In future research we will quantify the effects of this approximation.

We would also like to point out that in many cases (see below) the computation of the statistic can be done efficiently *within* the itemset generation procedure (*largeitems*) itself. This can be used to modify the algorithm to make it more efficient once a specific itemset generation procedure is used. This is the case if the function f that computes the statistic on transactions $T_1, T_2,\ldots, T_s$ is a recursive function on s, that is,

$$f(T_1, T_2,\ldots, T_s) = g(f(T_1, T_2,\ldots, T_{s-1}), f(T_s), s) \qquad (4.1)$$

Many statistics, such as *mean* and *market share*, are recursive. For example, $Mean(T_1, T_2,\ldots, T_s) = [Mean(T_1, T_2,\ldots, T_{s-1}) \times (s-1) + Mean(T_s)] / s$.

In this section we presented an algorithm *SigSQrules* for generating significant SQ rules. However, as mentioned in the introduction, for any given level of significance for a rule, the fact that thousands of rules are evaluated for their significance makes it possible to discover a certain number of false rules. This is the well known multiple hypothesis testing problem [4]. While it is difficult to eliminate this problem, it is possible to quantify this effect. In the next section we discuss the problem of false discovery in detail and present an algorithm for determining the false discovery rate associated with the discovery of significant SQ rules.

5. FALSE DISCOVERY OF SQ RULES

As mentioned above, when multiple rules are tested in parallel for significance, it is possible to learn a number of "false" rules by chance alone. Indeed, this is a problem for many rule discovery methods in the data mining literature. The *false discovery rate* (*FDR*) is the expected percentage of false rules among all the discovered rules. Prior work in statistics has taken two approaches to deal with the multiple hypothesis testing problem [4, 17, 34]. One approach attempts to lower the false discovery rate by adjusting the significance level at which each rule is tested. As we will describe below, this approach is not suitable for data mining since it will result in very few rules being discovered. The second approach assumes that a given number of false discoveries should be expected, and focuses on estimating what the false discovery rate (*FDR*) exactly is. This is more useful for data mining, since it permits the discovery of a reasonable number of rules, but at the same time computes a *FDR* that can give users an idea of what percentage of the discovered rules are spurious. In this section, we first review key ideas related to the multiple hypotheses testing problem and then present a nonparametric method to determine false discovery rate for our procedure.

For significance tests for a single rule, the significance level α is defined as the probability of discovering a significant rule when the LHS and RHS of the rule are actually independent of each other; in other words, α is the probability of a false (spurious) discovery. For example, on a random dataset where all attributes are independent, if we test 10,000 rules, then by definition of α, we expect 10,000 × 5% = 500 false discoveries by pure chance alone. When some of the attributes are dependent on each other, as is the case for most datasets on which rule discovery methods are used, the above approach cannot be used to get an expectation for the number of false rules. In such cases, two approaches are

possible. In statistics, a measure called familywise error rate (*FWER*) is defined as the probability of getting at least one false rule output. Most conventional approaches in statistics that deals with the multiple hypotheses testing problem use different methods to control *FWER* by lowering significance level for individual rule, α_{ind}. For example, Bonferroni-type procedures would have $\alpha_{ind} = \alpha$ / the number of rules tested, which is 5% / 10,000 = 5×10^{-6}. However, when the number of hypotheses tested is large (as is the case in data mining algorithms), extreme low α value, e.g. 5×10^{-6}, will result in very few rules discovered. The other type of approach, as taken recently in [4] estimates the false discovery rate (*FDR*), the expectation of the proportion of false discoveries in all discoveries.

Table 5.1 Confusion matrix of the number of rules

	Non-Significant Rules	Significant Rules
LHS independent of RHS	*a*	*b*
LHS dependent on RHS	*c*	*d*

In Table 5.1, the number of rules tested is ($a + b + c + d$), out of which ($a + b$) is the number of rules where the LHS of the rules is truly independent of the RHS, and ($c + d$) is the number of rules where there is a real relationship between the LHS and the RHS of the rules. The columns determine how many tested rules are output as significant or non-significant. The two terms *FDR* and *FWER* can be defined precisely as $FDR = Exp(b / b + d)$ and $FWER = Prob(b > 0)$.

We adopt *FDR* estimation in this section because it effectively estimates false discoveries without rejecting too many discovered rules. However, the method proposed in the literature [4, 7, 35] for *FDR* cannot be used for large scale rule discovery because of the following two reasons: first, the assumption that statistics of the rules tested are independent from each other (which some of the approaches use) is not true. For example, rules $A_1 = 1 \Rightarrow Mean(D_{A1=1})$ and $A_1 = 1 \wedge A_2 = 2 \Rightarrow Mean(D_{A1=1 \wedge A2=2})$ are not independent. In fact a large number of rules are related to each other in rule discovery because their LHS share common conditions and RHS come from the same attributes. Second, methods in statistics draw conclusions based on the number of rules tested (= $a + b + c + d$), however, as indicated in [25], *a* and *c* are unknown values due to the filtering by *support* constraint. Without making any assumptions, below we present another permutation-based method to estimate the *FDR* for our procedure for learning significant SQ rules.

Denote $N_{sig}(\alpha)$ to be the number of significant rules discovered from dataset *D* when the significant level = α. In Table 5.1, $N_{sig}(\alpha) = b + d$. Similar to the procedure described in Section 3, by keeping the values in attributes *A* intact and permuting the *B* attributes, we get a permutation dataset *D′*. Since we remove any relationship between *A* and *B* attributes by this procedure, all the LHS and RHS statistic of each rule tested in *D′* are independent. If we apply the significant rule discovery algorithm *SigSQrules*, the number of significant rules discovered from *D′* when the significant level = α will be one instance of false discovery, that is, $N_{sig\text{-}perm}(\alpha) = b$. It is easy to see that by creating multiple permutation datasets, we can estimate the expectation of the number of false discoveries and thus compute a false discovery rate $FDR = Exp(N_{sig\text{-}perm}(\alpha)) / N_{sig}(\alpha)$. We will describe the steps how $FDR(\alpha)$ can be estimated in detail in the Appendix.

In this section, we described the problem of multiple hypotheses testing and pointed out that for any given significance level a certain number of significant SQ rules will be discovered by chance alone. We then described an intuitive permutation based procedure to compute the false discovery rate. From a practical point of view the procedure described above can be used in conjunction with *SigSQrules* to discover a set of significant SQ rules and provide a number representing the percentage of these rules that are likely to be spurious.

6. EXPERIMENTS

In this section we present results from learning significant market share rules, a specific type of SQ rules. We started with user-level online purchase data gathered by comScore Networks, a market data vendor. The data consist of 100,000 users' online browsing and purchase behavior over a period of six months. The market data vendor tracks all online purchases explicitly by parsing the content of all pages delivered to each user. Starting from the raw data we created a dataset of purchases where each transaction represents a purchase made at an online retailer. Attributes of the transaction include user demographics, the site at which the purchase was made, the primary category (e.g. books, home electronics etc) of the site, the product purchased and the price paid for the product. Within a specific category, e.g. books, significant market share rules would be particularly interesting to discover. We selected many datasets with purchases belonging to each specific category and applied our method to learn several interesting significant market share rules. For space limitations we do not present all the results, but report the results for learning market share rules for the top three retailers in the online book industry. Specifically the dataset consists of all transactions in which a book was purchased at any site and we use the methods presented in the paper to learn market share rules for the top 3 sites – Amazon.com, Barnes&Noble and Ebay. The total number of transactions was 26,439 records and we limit the rules to having at most five items on the LHS of a rule.

6.1 Rule Examples

Among the most significant market share rules (as determined by the *p*-values of these rules), we picked four rules to list that were particularly interesting for each online retailer.

Amazon.com

(1) Education = High School ⇒ $\text{marketshare}_{\text{Amazon}}$ = 42.72%, support = 20.7%, CI = (46.07%, 50.92%)

(2) Region = West ∧ Household Size = 2 ⇒ $\text{marketshare}_{\text{Amazon}}$ = 57.93%, support = 7.9%, CI = (44.36%, 52.50%)

(3) Region = South ∧ Household Size = 4 ⇒ $\text{marketshare}_{\text{Amazon}}$ = 38.54%, support = 5.4%, CI = (43.76%, 53.39%)

(4) 35 < Household Eldest Age < 40 ∧ ISP = Broadband ⇒ $\text{marketshare}_{\text{Amazon}}$ = 60.52%, support = 4.3%, CI = (42.88%, 53.99%)

Barnesandnoble.com

(1) Education = Graduate ∧ Household Size = 2 ⇒ $\text{marketshare}_{\text{BN}}$ = 13.12%, support = 6.0%, CI = (16.81%, 25.68%)

(2) 50 < Household Eldest Age < 55 ∧ Income > 100K ⇒ $\text{marketshare}_{\text{BN}}$ = 30.28%, support = 4.2%, CI = (16.05%, 26.79%)

(3) Region = South ∧ Household Size = 3 ∧ Child = Yes ⇒ $\text{marketshare}_{\text{BN}}$ = 13.27%, support = 4.2%, CI = (16.68%, 26.10%)

(4) Region = South ∧ 60 < Household Eldest Age < 65 ⇒ $\text{marketshare}_{\text{BN}}$ = 39.84%, support = 2.8%, CI = (15.55%, 27.10%)

Ebay.com

(1) Education = College ∧ Region = South ⇒ $marketshare_{Ebay}$ = 8.28%, support = 6.9%, CI = (11.70%, 17.71%)
(2) Education = College ∧ Region = North Central ⇒ $marketshare_{Ebay}$ = 21.77%, support = 4.0%, CI = (11.05%, 18.29%)
(3) Region = South ∧ Income > 100K ⇒ $marketshare_{Ebay}$ = 4.83%, support = 2.9%, CI = (9.54%, 20.46%)
(4) 18 < Household Eldest Age < 20 ⇒ $marketshare_{Ebay}$ = 27.50%, support = 2.8%, CI = (10.12%, 19.70%)

Rule (4) for Amazon.com indicates that it is doing particularly well in households with middle-aged heads that have broadband access. The market share for Amazon.com in this segment lies significantly outside the confidence interval computed for the rule. On the other hand, rule (1) for Barnesandnoble.com shows that they are doing poorly selling to a segment which perhaps represents well educated couples. Given that this is a large segment (support = 6%), this rule suggests that they could try and examine why this is the case and how they can achieve greater penetration in this segment. In Ebay's case, all four rules are very interesting. Rule (4) indicates that they have high market share among teenagers, while rule (3) describes a segment they clearly have trouble penetrating. For many other categories too (travel and home electronics in particular) the significant SQ rules that we learned were highly interesting. As these examples suggest, these rules can be insightful, identify interesting segments and have significant business potential.

6.2 Varying support and significance levels

To test how the methods perform as the minimum support and significance levels vary, for one site we generated significant SQ rules for many values of the minimum support and significance level parameters. Figures 6.1 and 6.2 show how the number of significant rules and the false discovery rate vary with support.

As the minimum support threshold is lowered the number of significant SQ rules discovered increases. However the *FDR* increases as the support threshold is lowered, suggesting a tradeoff between discovering many significant rules while keeping the *FDR* low. A practical outcome is that it may be desirable to have higher minimum supports (to keep *FDR* low), but not too high that very few rules are discovered. Figures 6.3 and 6.4 illustrate a similar tradeoff for the significance level parameter. As α decreases *FDR* is lower, but this results in fewer number of significant rules being discovered. Again, the implication is that it may be desirable to have a low α (to keep *FDR* low) but not too low that very few rules are discovered.

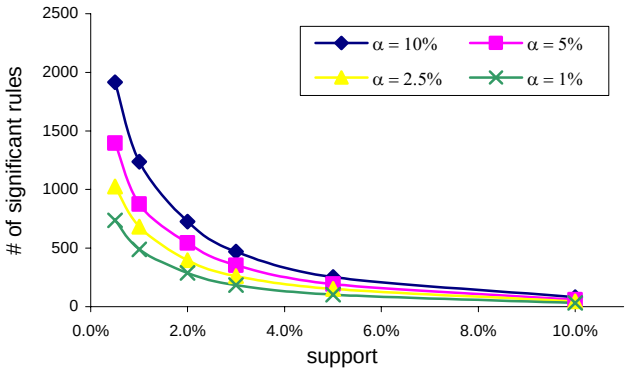

Figure 6.1. Effect of support on # of rules

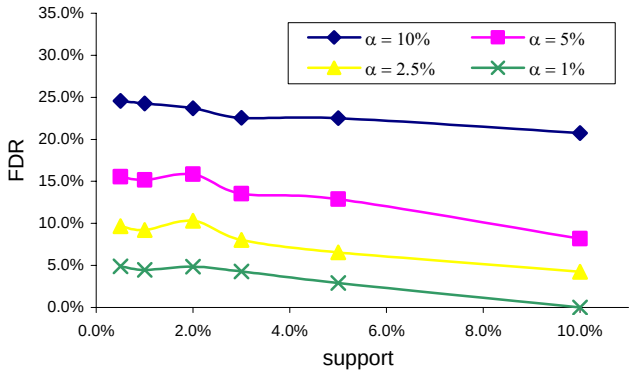

Figure 6.2. Effect of support on FDR

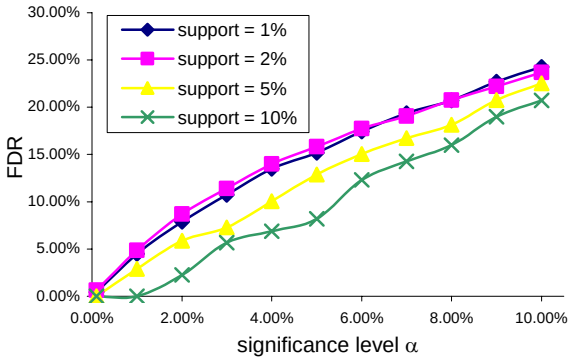

Figure 6.3. Effect of α on FDR

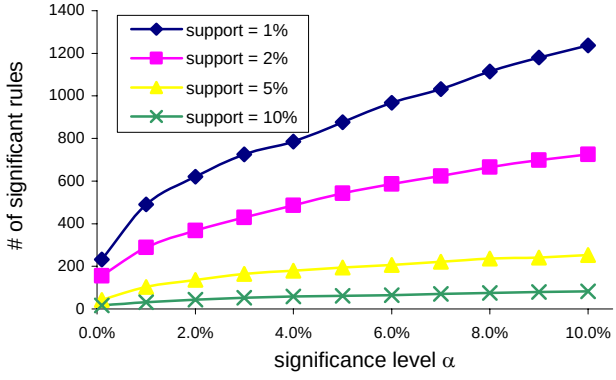

Figure 6.4. Effect of α on # of rules

6.3 Summary results for online book retailers

Based on this general tradeoff we chose minimum support of 2% and chose an α of 2.5% in order to report summary results for the three sites. Table 6.1 summarizes the number of significant rules discovered and the false discovery rates of the procedure. As the values in the table and the examples above show, our procedure can be used effectively to learn a good set of significant SQ rules while keeping the false discovery rates reasonable.

Table 6.1 Summary of results

Web site	Significant Rules	False Discovery Rate
Amazon	651	6.30%
Barnesandnoble	393	9.67%
Ebay	679	5.60%

In this section we first presented compelling examples of rules discovered that illustrate the potential of learning significant market share rules. We then examined how the number of significant rules discovered and the false discovery rate changes with the support and significance level (α) parameters. The results of this analysis suggested a tradeoff between generating significant rules and keeping the false discovery rate low. Based on this tradeoff we identified a specific value of the support and significance parameters and showed the number of rules discovered for these values.

7. RELATED WORK

We compare our work with the literature based on three aspects: rule structure, rule significance, and methodology.

Rule structure. Rule discovery methods on a quantitative dataset can be traced back to [29], where rules of the form $x_1 < A < x_2 \Rightarrow y_1 < B < y_2$ are discovered. [31] extends the structure to be conjunctions of multiple conditions on both antecedent and consequent of a rule, and proposes their discovery method based on the *Apriori* algorithm [1]. Although rules in [31] are important, partitions like $y_1 < B < y_2$ for continuous attributes on the RHS of a rule only gives partial description of the subset satisfying the LHS of the rule and partial descriptions sometimes are misleading. Observing this problem, [2] introduces a new structure where the consequent of a rule is $Mean(D_X)$ or $Variance(D_X)$ to summarize the behavior of the subset satisfying the antecedent. [33] further extends the form of the consequent of the rule, such that it can be of $Min(D_X)$, $Max(D_X)$, or $Sum(D_X)$. Our rule structure is based on prior work: the antecedent is conjunctions of conditions, while the consequent can be any aggregate function f on multiple attributes to describe the behavior of the subset satisfying the antecedent.

Rule significance. Any combination of attributes with conditions can potentially form a rule. Researchers use different measurements, e.g. support and confidence, to select only *important* rules from all possible rules. Based on the support and confidence framework, many metrics have been developed, such as gain [15], conviction [10], unexpectedness [27]. Although these metrics can be generalized to rules where the antecedent and consequent are both conjunctions of the form $value_1 < Attribute < value_2$ for quantitative datasets, they are not applicable for rules whose consequent is a function, such as $Mean(D_X)$, or in general, $f(D_X)$. To solve this non-trivial problem, we use statistical significance tests to evaluate rules, so that the consequent of a rule is not expected by chance alone. In the data mining literature, statistical significance tests are commonly used in many applications. For example, chi-square (χ^2) is a statistic to test correlations between attributes in binary or categorical data, and it has been applied to discover correlation rules [9], actionable rules [23], and contrast sets [3, 32]. For sparse data, [35, 36] employ Fisher's Exact Test to detect anomaly patterns for disease outbreaks. As mentioned in Section 3, these two tests are special cases of our significance test when we apply our significance definition to categorical data. For quantitative rules in [2], the authors use a standard Z-test to determine the significance of inequality of means between a subset D_X and its complement $D - D_X$. [33] defines a new measurement, *impact*, to evaluate quantitative rules, where *impact* can identify those groups that contribute most to some outcome, such as profits or costs. For areas other than rule discovery, standard Z-tests with log-linear models is used in Exploratory Data Analysis for OLAP data cubes [30]. Our significance test is different from the above primarily because (i) our significance definition is applicable to any user-defined aggregate function $f(D_X)$, and (ii) we using nonparametric methods to construct distributions and confidence intervals, in which $f(D_X)$ is expected from random effects alone.

Methodology. Nonparametric statistics is philosophically related to data mining, in that both methods typically make no assumptions on distributions of data or test statistics. Even with known distribution of a statistic, nonparametric methods are useful to estimate parameters of the distribution [13]. Nonparametric methods are widely used on testing models that are built from data: as earliest in [18], the author uses randomization tests to tackle a model overfitting problem; [20] compares bootstrap and cross-validation for model accuracy estimation; for decision trees, [14, 26] use permutation tests to select attributes based on 2×2 contingency tables. Rule discovery is to learn local features, which is inherently different from models. Although we have seen methods using parametric hypothesis testing approach to learning rules from dataset [5, 6], no prior work has been found on discovering large number of rules based on nonparametric significance tests.

The problem of multiple hypothesis testing/multiple comparison is well known in rule discovery, a good review of which can be found in [19]. On sparse binary data, [25] shows that with proper support and confidence control, very few false rules will be discovered. However, rule discovery on quantitative data faces much more complicated challenges, and conventional *p*-value adjustment methods cannot be directly applied. To solve this problem, we employ false discovery rate [4] metric to estimate the number of false rules discovered due to testing a large number of rules. In data mining, FDR has been shown useful in [7, 36] for categorical data with known number of hypotheses, and we extend it to quantitative rules with resampling methods.

8. CONCLUSION

In this paper we defined a new category of rules, SQ rules, and the significance of SQ rules, on quantitative data. Then we presented a permutation-based algorithm for learning significant SQ rules. Furthermore, we show how an explicit false discovery rate can be estimated for our procedure, which makes the approach useful from a practical perspective. We presented experiments in which we discovered market share rules, a specific

type of SQ rules, in real online purchase datasets and demonstrated that our approach can be used to learn interesting rules from data.

We would also like to point out that it is possible to compute the false discovery rate (*FDR*) for several possible significance levels in an efficient manner (without creating permutation datasets for each significance level). Although a detailed presentation of this is beyond the scope of this paper, in the appendix we provide an overview of how this can be done. One main advantage of being able to do this is that significant SQ rules can be discovered at a chosen significance level that is computed from some desired *FDR*. Hence rather than just *estimating FDR* we may be able to discover significant rules *given* a specific *FDR*. However this needs to be studied in greater detail in future work.

9. REFERENCES

[1] Agrawal, R. and Srikant, R., Fast Algorithms for Mining Association Rules, in *Proceedings of the 20th International Conference on Very Large Databases*, Santiago, Chile, 1994.

[2] Aumann, Y. and Lindell, Y., A Statistical Theory for Quantitative Association Rules, in *Proceedings of The Fifth ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pp. 261-270, San Diego, CA, 1999.

[3] Bay, S. D. and Pazzani, M. J., Detecting Change in Categorical Data: Mining Contrast Sets, in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 302 - 306, San Diego, CA, 1999.

[4] Benjamini, Y. and Hochberg, Y., Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing, *Journal of Royal Statistical Society B*, vol. 57, iss. 1, pp. 289-300, 1995.

[5] Bolton, R. and Adams, N., An Iterative Hypothesis-Testing Strategy for Pattern Discovery, in *Proceedings of the Ninth ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pp. 49-58, Washington, DC, 2003.

[6] Bolton, R. J. and Hand, D. J., Significance Tests for Patterns in Continuous Data, in *Proceedings of the 2001 IEEE International Conference on Data Mining*, pp. 67-74, San Jose, CA, 2001.

[7] Bolton, R. J., Hand, D. J., and Adams, N. M., Determining Hit Rate in Pattern Search, in *Pattern Detection and Discovery, ESF Exploratory Workshop*, pp. 36-48, London, UK, 2002.

[8] Brijs, T., Swinnen, G., Vanhoof, K., and Wets, G., Using Association Rules for Product Assortment: Decisions Case Study, in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 254-260, San Diego, CA, 1999.

[9] Brin, S., Motwani, R., and Silverstein, C., Beyond Market Baskets: Generalizing Association Rules to Correlations, in *Proceedings of the ACM SIGMOD/PODS '97 Joint Conference*, pp. 265-276, Tucson, AZ, 1997.

[10] Brin, S., Motwani, R., Ullman, J. D., and Tsur, S., Dynamic Itemset Counting and Implication Rules for Market Basket Data, in *Proceedings ACM SIGMOD International Conference on Management of Data (SIGMOD'97)*, pp. 255-264, Tucson, AZ, 1997.

[11] Clark, P. and Niblett, T., The Cn2 Induction Algorithm, *Machine Learning*, vol. 3, pp. 261-283, 1989.

[12] Clearwater, S. and Provost, F., Rl4: A Tool for Knowledge-Based Induction, in *Procs. of the Second International IEEE Conference on Tools for Artificial Intelligence*, pp. 24-30, 1990.

[13] Efron, B. and Tibshirani, R. J., *An Introduction to the Bootstrap*. New York, NY: Chapman & Hall, 1993.

[14] Frank, E. and Witten, I. H., Using a Permutation Test for Attribute Selection in Decision Trees, in *Proceedings of 15th Int'l Conference on Machine Learning*, pp. 152-160, 1998.

[15] Fukuda, T., Morimoto, Y., Morishita, S., and Tokuyama, T., Data Mining Using Two-Dimensional Optimized Association Rules: Scheme, Algorithms and Visualization, in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (SIGMOD'96)*, pp. 13-23, Montreal, Quebec, Canada, 1996.

[16] Good, P., *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses - 2nd Edition*. New York: Springer, 2000.

[17] Hsu, J. C., *Multiple Comparisons - Theory and Methods*. London, UK: Chapman & Hall, 1996.

[18] Jensen, D., Knowledge Discovery through Induction with Randomization Testing, in *Proceedings of the 1991 Knowledge Discovery in Databases Workshop*, pp. 148-159, Menlo Park, 1991.

[19] Jensen, D. and Cohen, P. R., Multiple Comparisons in Induction Algorithms, *Machine Learning*, vol. 38, pp. 309-338, 2000.

[20] Kohavi, R., A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 1137-1143, San Mateo, CA, 1995.

[21] Lee, Y., Buchanan, B. G., and Aronis, J. M., Knowledge-Based Learning in Exploratory Science: Learning Rules to Predict Rodent Carcinogenicity, *Machine Learning*, vol. 30, pp. 217-240, 1998.

[22] Ling, C. X. and Li, C., Data Mining for Direct Marketing: Problems and Solutions, in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 73-79, New York, NY, 1998.

[23] Liu, B., Hsu, W., and Ma, Y., Identifying Non-Actionable Association Rules, in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 329-334, San Francisco, CA, 2001.

[24] Mani, D. R., Drew, J., Betz, A., and Datta, P., Statistics and Data Mining Techniques for Lifetime Value Modeling, in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 94-103, San Diego, CA, 1999.

[25] Megiddo, N. and Srikant, R., Discovering Predictive Association Rules, in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 274-278, New York, NY, 1998.

[26] Oates, T. and Jensen, D., Large Datasets Lead to Overly Complex Models: An Explanation and a Solution, in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 294-298, Menlo Park, CA, 1998.

[27] Padmanabhan, B. and Tuzhilin, A., A Belief-Driven Method for Discovering Unexpected Patterns, in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 94-100, New York, NY, 1998.

[28] Padmanabhan, B. and Tuzhilin, A., Small Is Beautiful: Discovering the Minimal Set of Unexpected Patterns, in *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 54-63, Boston, MA, 2000.

[29] Piatesky-Shapiro, G., Discovery, Analysis, and Presentation of Strong Rules, in *Knowledge Discovery in Databases*, Piatesky-Shapiro, G. and Frawley, W. J., Eds. Menlo Park, CA: AAAI/MIT Press, pp. 229-248, 1991.

[30] Sarawagi, S., Agrawal, R., and Megiddo, N., Discovery-Driven Exploration of Olap Data Cubes, in *Proceedings of the Sixth International Conference on Extending Database Technology (EDBT'98)*, pp. 168-182, Valencia, Spain, 1998.

[31] Srikant, R. and Agrawal, R., Mining Quantitative Association Rules in Large Relational Tables, in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, 1996.

[32] Webb, G., Butler, S., and Newlands, D., On Detecting Differences between Groups, in *Proceedings of the Ninth ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pp. 256-265, Washington, DC, 2003.

[33] Webb, G. I., Discovering Associations with Numeric Variables, in *Proceedings of The Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, 2001.

[34] Westfall, P. H. and Young, S. S., *Resampling-Based Multiple Testing - Examples and Methods for P-Value Adjustment*. New York, NY: John Wiley & Sons, Inc, 1993.

[35] Wong, W.-K., Moore, A., Cooper, G., and Wagner, M., Rule-Based Anomaly Pattern Detection for Detecting Disease Outbreaks, in *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, Edmonton, Canada, 2002.

[36] Wong, W.-K., Moore, A., Cooper, G., and Wagner, M., Bayesian Network Anomaly Pattern Detection for Disease Outbreaks, in *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Washington, DC, 2003.

APPENDIX: Discovering false discovery rates for multiple significance levels

Let us continue to use the example $N_{perm} = 999$ and $\alpha = 5\%$. On the dataset *D*, from the algorithm *SigSQrules* we generate significant rules as well as each rule's *p*-value. Because there are N_{perm} values in each distribution, the smallest possible *p*-value from the permutation tests is $1/(N_{perm}+1) = 0.001$, and all possible *p*-values are $S = \{ 1/(N_{perm}+1) = 0.001, 2/(N_{perm}+1) = 0.002, \ldots \alpha = 0.05 \}$. Let $N_{sig}[\alpha_{ind}]$ be the number of significant rules whose *p*-value is no larger than $\alpha_{ind} \in S$. For example, if there are 50 rules whose *p*-value = 0.001, and 30 rules whose *p*-value = 0.002, then $N_{sig}[0.001] = 50$ and $N_{sig}[0.002] = 50 + 30 = 80$. Without further permutation tests, with $N_{sig}[]$ we know how many rules will be discovered if we lower the significance level from α to α_{ind}. For example, if $\alpha_{ind} = 0.002$, there are only $N_{sig}[0.002] = 80$ rules whose *p*-value is no larger than $\alpha_{ind} = 0.002$, therefore we expect to discover 80 rules.

Similarly, for each permutation dataset *D'*, at each significance level $\alpha_{ind} < \alpha$ we can compute the number of significant rules and their *p*-values by applying *SigSQrules* only once. Note that all discoveries from *D'* are false discoveries, because the relationships between *A* and *B* are removed. Let $N_{sig\text{-}perm}[i][\alpha_{ind}]$ be the number of discoveries from permutation datasets *D'*[*i*]. For example, $N_{sig\text{-}perm}[1][0.002] = 20$ means we have 20 discoveries from the permutation dataset *D'*[1] at $\alpha_{ind} = 0.002$. We implement this procedure on multiple permutation datasets, and *Median*($N_{sig\text{-}perm}[][\alpha_{ind}]$) is the estimate of false discoveries at each significance level α_{ind} on permutation datasets. Therefore, $FDR(\alpha_{ind}) = Median(N_{sig\text{-}perm}[][\alpha_{ind}]) / N_{sig}[\alpha_{ind}]$. We use *Median* to estimate the expectation, which conforms to nonparametric statistical considerations (median is the best estimator for expectation when the underlying distribution is unknown).

Empirically, we showed in Figure 6.3 that $FDR(\alpha_{ind})$ is an increasing function on α_{ind}. It means that by decreasing α_{ind}, we can control $FDR(\alpha_{ind})$ to a smaller value. We are not always guaranteed, though, to be able to set an individual significance level such that *FDR* < 5%. It is possible that even when we decrease α_{ind} to a level that almost no rules are discovered, *FDR* is still much larger than 5%. In other words, there are always a large proportion of spurious rules discovered from some datasets. For example, if attributes independent based on a test *statistic*, then $Median(N_{sig\text{-}perm}[][\alpha_{ind}]) \approx N_{sig}[\alpha_{ind}]$ for all significance levels, and $FDR \approx 1$. We want to point out that this is a desirable property of our method on controlling *FDR*, because there are many real-world datasets whose attributes are truly independent from each other. Traditional methods cannot estimate how many rules should be discovered, but with our technique, we can draw the conclusion that, there is no rule to be discovered because none of the rules is better than chance. This nonparametric method to estimate and control *FDR* is applicable to quantitative datasets and broad types of rules.