Supporting Exploratory Hypothesis Testing and Analysis

GUIMEI LIU, Institute for Infocomm Research
HAOJUN ZHANG, University of Wisconsin–Madison
MENGLING FENG, Institute for Infocomm Research
LIMSOON WONG, National University of Singapore
SEE-KIONG NG, Institute for Infocomm Research

Conventional hypothesis testing is carried out in a hypothesis-driven manner. A scientist must first formulate a hypothesis based on what he or she sees and then devise a variety of experiments to test it. Given the rapid growth of data, it has become virtually impossible for a person to manually inspect all data to find all of the interesting hypotheses for testing. In this article, we propose and develop a data-driven framework for automatic hypothesis testing and analysis. We define a hypothesis as a comparison between two or more subpopulations. We find subpopulations for comparison using frequent pattern mining techniques and then pair them up for statistical hypothesis testing. We also generate additional information for further analysis of the hypotheses that are deemed significant. The number of hypotheses generated can be very large, and many of them are very similar. We develop algorithms to remove redundant hypotheses and present a succinct set of significant hypotheses to users. We conducted a set of experiments to show the efficiency and effectiveness of the proposed algorithms. The results show that our system can help users (1) identify significant hypotheses efficiently, (2) isolate the reasons behind significant hypotheses efficiently, and (3) find confounding factors that form Simpson's paradoxes with discovered significant hypotheses.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Exploratory hypothesis testing, comparative data analysis, actionable knowledge, exploratory data mining

ACM Reference Format:
Guimei Liu, Haojun Zhang, Mengling Feng, Limsoon Wong, and See-Kiong Ng. 2015. Supporting exploratory hypothesis testing and analysis. ACM Trans. Knowl. Discov. Data 9, 4, Article 31 (June 2015), 24 pages.
DOI: http://dx.doi.org/10.1145/2701430

1. INTRODUCTION

Hypothesis testing is a statistical procedure for testing whether chance is a plausible explanation of an experimental finding. It enables scientists to distinguish findings that represent systematic effects in the data from those that are due to random chance. Hypothesis testing involves a comparison of two or more subpopulations. One example

This work is supported in part by the Singapore Agency for Science, Technology, and Research grant SERC 102 1010 0030. The main work was done when G. Liu and H. Zhang worked at the National University of Singapore.

Authors' addresses: G. Liu, M. Feng, and S.-K. Ng, Data Analytics Department, Institute for Infocomm Research, 1 Fusionopolis Way, Singapore, 138632; emails: {liug, mfeng, skng}@i2r.a-star.edu.sg; H. Zhang, Department of Computer Sciences, University of Wisconsin–Madison, 1210 W. Dayton St, Madison WI 53706, USA; email: hzhang0418@cs.wisc.edu; L. Wong, Department of Computer Science, National University of Singapore, COM1, 13 Computing Drive, Singapore 117417; email: wongls@comp.nus.edu.sg.

DOI: http://dx.doi.org/10.1145/2701430

hypothesis is “Smokers are more vulnerable to the H1N1 flu than nonsmokers.” To test this hypothesis, we need to compare the occurrence of H1N1 flu infection between two subpopulations: in this case, smokers and nonsmokers. The outcome of hypothesis testing can help people make decisions. For example, by knowing which group of people is more vulnerable to a certain type of flu, doctors can vaccinate this group of people first to prevent the spread of the flu.

Conventional hypothesis testing is usually carried out in a *hypothesis-driven* manner. A scientist must first formulate a hypothesis based on what he or she sees from the data and his or her knowledge and then devise a variety of experiments to test it. This presents a possible *catch-22* situation, for it is often the case that people want to find something from their data but do not know what to find. Even if a person has much domain knowledge and ample experience, the data may still contain something useful of which he or she is not aware. For example, even an experienced doctor may not know all of the risk factors of a complex disease.

With the rapid development of information technology, more and more data have been accumulated and stored in digital format. These data provide rich sources for making new discoveries. However, the sheer volume of the data available today makes it impossible for people to inspect all of the data manually. As a result, lots of useful knowledge may go undiscovered. This calls for the need to develop a system for automatic hypothesis testing in a *data-driven* manner.

Data mining is an important tool to transform data into knowledge in a data-driven manner. Data mining does not start from a preconception or a specific question such as hypothesis testing. Instead, it aims to automatically extract useful information from large volumes of data via exploratory search. It is able to detect things that are hard to detect manually or are overlooked by users. In our previous work [Liu et al. 2011], we have formally defined the exploratory hypothesis testing and analysis problem and have proposed algorithms for solving the problem by building on and extending existing data mining techniques. More specifically, given a dataset, we formulate and test tentative hypotheses based on the attributes in the dataset, the nature of the attributes, and the statistics of the data. The space of all possible hypotheses can be very large. We employ techniques developed for frequent pattern mining to efficiently explore the space of tentative hypotheses. In many cases, it is not sufficient to just know whether a hypothesis is statistically significant. It is more interesting and important to know the reasons behind significant hypotheses. We provide tools for users to analyze the significant hypotheses and identify factors that contribute to the difference. Another reason for the need for further analysis is that some of the significant hypotheses generated may be spurious, because the exploration is not guided by domain knowledge. Our analyzing tools can detect confounding factors that may lead to spurious hypotheses.

In this article, we further address the problem of removing redundancy from the generated hypotheses, as the number of significant hypotheses generated on some datasets can be overwhelmingly large. It is infeasible to inspect and analyze all of them. We have observed that many of the generated hypotheses are very similar to one another. To reduce users’ burden, we develop algorithms to find a succinct set of representative hypotheses for users to explore.

The rest of the article is organized as follows. Section 2 formally defines the problem. In Section 3, we develop efficient algorithms for automatic hypothesis testing and analysis by using and extending well-established frequent pattern mining techniques. Experiment results are reported in Section 4. Section 5 describes related work. Finally, Section 6 concludes the article.

2. PROBLEM DEFINITION

Hypothesis testing is a test of significance on a difference. A difference is *statistically significant* if it is unlikely to have occurred by chance. Hypothesis testing can be

Table I. An Example Dataset

PID	Race	Gender	Age	Smoke	Stage	Treatment	Response
1	Caucasian	M	45	Yes	1	A	Positive
2	Asian	M	40	No	1	A	Positive
3	African	F	50	Yes	2	B	Negative
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
N	Caucasian	M	60	No	2	B	Negative

conducted on n subpopulations where $n \geq 1$. Hypothesis testing involving only one subpopulation compares the statistics of one subpopulation with the parameters of the general population. The well-studied association rule mining problem can be viewed as hypothesis testing involving only one subpopulation. The results of hypothesis testing involving more than two subpopulations are usually hard to interpret. It is not easy for users to tell which subpopulation contributes the most to the difference. In the end, users still need to resort to pairwise comparisons to get a clear picture. Hence, in this article, we focus on the case when $n = 2$.

The hypothesis testing process consists of four main steps:

(1) State the relevant null and alternative hypotheses. The null hypothesis is always “There is no difference.” The alternative hypothesis is “There is difference.”
(2) Choose a proper statistical test based on the type and distribution of the attribute on which the subpopulations are compared. An introduction to the statistical tests and when they should be used can be found in Motulsky [1995].
(3) Calculate the p-value using the chosen test. The p-value is the probability of obtaining a statistic at least as extreme as the one that was actually observed, given that the null hypothesis is true. The lower the p-value, the less likely that the observed difference is due to random chance and thus the more statistically significant the difference.
(4) Decide whether to reject or accept the null hypothesis based on the p-value. Conventionally, a p-value of 0.05 is recognized as low enough to reject the null hypothesis and accept the alternative hypothesis.

In the rest of this section, we describe how to automate the testing process to test all possible hypotheses in a given dataset. We define the problem in the situation when the target attribute is categorical. The target attribute is the attribute on which the significance of the difference is tested. It is straightforward to generalize the definitions to the situation when the target attribute is continuous.

2.1. Hypothesis Formulation

We use the example dataset shown in Table I to illustrate how to formulate hypotheses. Each row in the dataset is the medical record of a patient. The last column is the response of patients to a certain treatment. One example hypothesis is “Treatment A is more effective than treatment B.” The two subpopulations under comparison are “patients undergone treatment A” and “patients undergone treatment B,” and they are compared on the attribute “Response.” There are two types of attributes in a hypothesis. One type of attributes, such as “Treatment,” is used to define subpopulations; we call these *grouping attributes*. A grouping attribute cannot be continuous. The other type of attributes, such as “Response,” is the attributes on which the significance of the difference is tested; we call these *target attributes*.

Given a dataset, we ask domain users to specify a target attribute as the objective of the exploratory hypothesis testing. If the target attribute is a categorical attribute, we further ask domain users to choose a value that is the most interesting to them; we call this value the *target attribute value*. We also ask domain users to specify the grouping attributes if possible (if not, we simply use all categorical attributes in the given dataset as grouping attributes). This should be easy for domain users, as they usually have some rough ideas on which attributes are of interest for comparison and which attributes can be used for grouping.

A subpopulation is defined by a set of attribute-value pairs. We call an attribute-value pair an *item* and a set of items a *pattern*.

Definition 2.1 (*Pattern*). A pattern is a set of attribute-value pairs (items), denoted as $P = \{A_1 = v_1, A_2 = v_2, \ldots, A_k = v_k\}$, where A_i is an attribute ($1 \leq i \leq k$), v_i is a value taken by attribute A_i, and $A_i \neq A_j$ if $i \neq j$.

If an attribute-value pair $A = v$ appears in a pattern P, we say that P *contains* attribute A. Each pattern defines a subpopulation. For example, pattern {EthnicGroup=Caucasian, Gender=Male} defines the subpopulation consisting of Caucasian male patients. In the rest of this article, we use $T(P)$ to denote the subpopulation defined by pattern P. The support of a pattern P, denoted as $sup(P)$, is defined as the number of records containing P. The support of pattern P is the same as the number of records in $T(P)$.

Given two patterns P and P', if every item in P is also in P', then P is called a *subpattern* of P', denoted as $P \subseteq P'$, and P' is called a *superpattern* of P, denoted as $P' \supseteq P$. If P is a subpattern of P' and P has one less item than P', then we call P an *immediate subpattern* of P' and P' an *immediate superpattern* of P. If $P \subseteq P'$, then we have $T(P) \supseteq T(P')$.

In hypothesis testing, users usually study one factor at a time. Hence, in this work, we require that the defining patterns of two subpopulations under comparison differ by one and only one item. For example, comparing subgroup {EthnicGroup=Caucasian, Gender=Male} with subgroup {EthnicGroup=Caucasian, Gender=Female} is acceptable, whereas comparing subgroup {EthnicGroup=Caucasian, Gender=Male} with subgroup {EthnicGroup=Asian, Gender=Female} is less intuitive because even if the difference between the two subpopulations is statistically significant, it is not easy for users to conjecture which attribute contributes to the difference. Now we formally define tentative hypotheses.

Definition 2.2 (*Tentative Hypothesis*). Let A_{target} be a categorical target attribute, v_{target} be the target attribute value, and P_1 and P_2 be two patterns that contain the same set of attributes and differ by one and only one item, denoted as $P_1 = P \cup \{A_{diff} = v_1\}$, $P_2 = P \cup \{A_{diff} = v_2\}$. The tentative hypothesis on the two subpopulations defined by P_1 and P_2 is represented as $H = \langle P, A_{diff} = v_1|v_2, A_{target}, v_{target}\rangle$. Pattern P is called the *context* of H, attribute A_{diff} is called the *comparing attribute* of H, and P_1 and P_2 are called the *two subpopulations* of H. The null hypothesis is $p_1 = p_2$, and the alternative hypothesis is $p_1 \neq p_2$, where p_i is the proportion of v_{target} in P_i—that is, $p_i = \frac{sup(P_i \cup \{A_{target}=v_{target}\})}{sup(P_i)}$, $i = 1, 2$.

Based on the definition, the hypothesis "Treatment A is more effective than treatment B" can be represented as $\langle\{\}, Treatment = A|B, Response, positive\rangle$.

2.2. Choosing a Proper Test and Calculating the *p*-Value

The selection of a proper statistical test depends on the type and distribution of the target attribute. Most of the statistical tests can be integrated into our system seamlessly. Given a statistical test, we need to scan the dataset and collect some statistics for the calculation of the *p*-value.

For the example hypothesis H: "Treatment A is more effective than treatment B," the two subpopulations are compared on attribute "Response," and it is nominal; hence,

Table II. Statistics Needed for Calculating the p-Value of $H = \langle \{\}, \textit{Treatment} = A|B, \textit{Response}, \textit{positive} \rangle$ Using the χ^2-Test

Patterns	Support	p_i
Treatment=A	1,000 (n_1)	$p_1 = 89\%$
Treatment=A, Response=positive	890	
Treatment=B	1,000 (n_2)	$p_2 = 83\%$
Treatment=B, Response=positive	830	

we choose the χ^2-test with Yates' correction [Yates 1934] to calculate the p-value. The χ^2-test is a test for examining the association between two categorical attributes, and it requires four statistics to be collected: the size of the two subpopulations under comparison, denoted as n_1 and n_2, and the proportion of the target attribute value in the two subpopulations, denoted as p_1 and p_2. The four values of H are shown in Table II. Given the four values, the χ^2-score is calculated as follows:

$$\chi^2_{Yates} = \frac{(n_1 + n_2)(n_1 n_2 |p_1 - p_2| - (n_1 + n_2)/2)^2}{n_1 n_2 (n_1 p_1 + n_2 p_2)(n_1 + n_2 - n_1 p_1 - n_2 p_2)}. \tag{1}$$

The χ^2-score of H is 14.46, and the degree of freedom is 1. We can get the corresponding p-value by looking up an χ^2 distribution table. The p-value of H is around 0.00014.

2.3. Deciding the Statistical Significance of a Hypothesis

Conventionally, a p-value of 0.05 is recognized as low enough to reject the null hypothesis if one single hypothesis is tested. A p-value of 0.05 means that there is a 0.05 probability that the null hypothesis is true but we are wrongly rejecting it. If we test 1,000 random hypotheses at the significance level of 0.05, then around 50 hypotheses will be regarded as significant just by random chance. Such hypotheses are false positives. Here we are testing large numbers of hypotheses simultaneously, so we need to control the number of false positives. We have conducted a comprehensive study on several multiple testing correction methods for controlling false positives in association rule mining [Liu et al. 2011]. Here we use Bonferroni correction [Abdi 2007] and Benjamini and Hochberg's method [Benjamini and Hochberg 1995] for their simplicity and efficiency.

Bonferroni correction [Abdi 2007] is one of the most commonly used approaches for multiple testing correction. It aims at controlling the family wise error rate (FWER)—the probability of making at least one false discovery among all hypotheses. The basic idea is that if we test n hypotheses, then one way of maintaining the FWER is to test each individual hypothesis at a statistical significance level of $1/n$ times what it would be if only one hypothesis were tested. Bonferroni correction is computationally simple, but it can be very conservative and may inflate the rate of false negatives unnecessarily. To use Bonferroni correction, we count the total number of tests being performed during the hypothesis generation process, denoted as n, and then use max_pvalue/n to replace max_pvalue, where max_pvalue is the statistical significance threshold specified by users.

Benjamini and Hochberg's method [Benjamini and Hochberg 1995] controls the false discovery rate (FDR)—the expected proportion of false positives, which is less stringent than FWER. Let $H_1, H_2, \ldots, H_n$ be the n hypotheses tested; they are sorted in ascending order of p-value. Their corresponding p-values are $p_1, p_2, \ldots, p_n$. To control FDR at a level of q, we get the largest i, denoted as k, for which $p_i \leq \frac{i}{n} q$, and then regard all H_i, $i = 1, 2, \ldots, k$, as statistically significant.

Statistical significance versus domain significance. Sometimes a statistically significant result can have little or no domain significance. For example, given large sample sizes, a difference in 5 beats per minute in the pulse rate in a clinical trial involving two drugs can give a statistically significant difference, whereas the average difference may hardly bring about a drastic metabolic change between the two groups, because domain significance mainly depends on the difference between the two means or proportions, whereas statistical significance also depends on standard error. If standard error is small enough, a difference can be statistically significant even if it is not domain significant. Only domain users can decide the level at which a result is regarded as domain significant.

2.4. Representative Hypotheses

The number of significant hypotheses generated on a dataset can be very large, which imposes a great challenge on visualizing, exploring, and further analyzing the generated hypotheses. We have observed that many of the hypotheses are very similar to one another. Given two hypotheses H and H' such that H' and H have the same comparing items and the context of H' is more specific than that of H, if the difference between the two subpopulations of H' is of the same direction as, and is also close to, that of H, then H' is not of much interest if we already have H. It is desirable to remove such redundant hypotheses and find a small number of hypotheses to represent all significant hypotheses.

Definition 2.3 (*ϵ-cover*). Given two hypotheses $H = \langle P, A_{diff} = v_1|v_2, A_{target}, v_{target}\rangle$ and $H' = \langle P', A_{diff} = v_1|v_2, A_{target}, v_{target}\rangle$, if $T(P') \subseteq T(P)$, $(p'_1 - p'_2)(p_1 - p_2) \geq 0$, and $p'_1 - p'_2 \leq (1 + \epsilon) \cdot (p_1 - p_2)$, where p_is and p'_is are the proportion of v_{target} in the two subpopulations of H and H', respectively, then H ϵ-covers H'.

Obviously, a hypothesis 0-covers itself. Hypotheses with different comparing items or different directions of difference cannot cover each other based on the definition. If the subpopulations defined by the context patterns of two hypotheses do not have subset-superset relationship, they cannot cover each other either. Our goal is to find a minimal set of hypotheses that ϵ-covers all significant hypotheses, and these hypotheses are called *representative hypotheses*. They can be viewed as a summarization of all significant hypotheses. The value of ϵ represents a trade-off between the size of the summarization and the information loss caused by the summarization. A smaller ϵ means a larger number of representative hypotheses and less information loss.

LEMMA 2.4. *Given two hypotheses $H = \langle P, A_{diff} = v_1|v_2, A_{target}, v_{target}\rangle$ and $H' = \langle P', A_{diff} = v_1|v_2, A_{target}, v_{target}\rangle$, if $T(P') = T(P)$, then H and H' ϵ-cover each other.*

PROOF. Given $T(P') = T(P)$, we have $T(P \cup \{A_{diff} = v_i\}) = T(P) \cap T(\{A_{diff} = v_i\}) = T(P') \cap T(\{A_{diff} = v_i\}) = T(P' \cup \{A_{diff} = v_i\})$, $i = 1, 2$. Consequently, we have $p_i = p'_i$, $i = 1, 2$. Therefore, H and H' ϵ-cover each other based on Definition 2.3. □

The preceding lemma indicates that if two patterns occur in the same set of records, then the hypotheses having them as context patterns are redundant to each other. We need to keep only one of them as a context pattern. In our implementation, we choose to use only frequent closed patterns as context patterns. A pattern is *closed* if it is the longest pattern among all patterns occurring in exactly the same set of records [Pasquier et al. 1999].

LEMMA 2.5. *Given two hypotheses $H = \langle P, A_{diff} = v_1|v_2, A_{target}, v_{target}\rangle$ and $H' = \langle P', A_{diff} = v_1|v_2, A_{target}, v_{target}\rangle$, if patterns P and P' are both closed, $P \neq P'$ and H' ϵ-covers H, then $P' \subset P$.*

The preceding lemma is straightforward to prove by using proof by contradiction. It implies that if we use only closed patterns as context patterns, then a necessary condition for a hypothesis H' to ϵ-cover H is that the context pattern of H' is a

Table III. Example of the Interesting Attribute "Time-of-Failure"

Pairs of Subpopulations	Failure Rates

model=A
model=B
	4%
	2%

model=A, time-of-failure=loading
model=B, time-of-failure=loading
	6.0%
	1.9%

model=A, time-of-failure=in-operation
model=B, time-of-failure=in-operation
	2.1%
	2.1%

model=A, time-of-failure=outputting
model=B, time-of-failure=outputting
	2.0%
	1.9%

Note: "Loading" is of particular interest because the two models show a very big difference.

Table IV. Example of a Spurious Hypothesis

Pairs of	Response		Proportion of	
Subpopulations	Positive	Negative	Positive Response	p-Value
Treatment=A	890	110	89.0%	0.0001
Treatment=B	830	170	83.0%	
Treatment=A, Stage=1	800	80	90.9%	0.0807
Treatment=B, Stage=1	190	10	95%	
Treatment=A, Stage=2	90	30	75%	0.2542
Treatment=B, Stage=2	640	160	80%	

Note: Attribute "stage" is a confounding factor.

subpattern of that of H. We utilize this necessary condition to generate representative hypotheses.

2.5. Hypothesis Analysis

In many cases, we are not only interested in knowing whether the difference between two subpopulations is significant; we are more interested in the reasons behind the difference. For example, given that the failure rate of one model of a product is significantly higher than that of another model, it is important for engineers to know under what situations the first model is more likely to fail so that they can improve its design accordingly. Table III compares the failure rate of two models of the same product. Model A has a higher failure rate than model B in general. However, after the two subpopulations are further divided using attribute "time-of-failure," we find that model A has comparable failure rate with model B at the time of "in-operation" and "outputting" but has an exceptionally high failure rate in the "loading" phase. This information is very useful, as it helps engineers narrow down the problem.

Another reason for the need for further analysis of significant hypotheses is that some of the significant hypotheses generated may be spurious since the search is exploratory in nature. For example, hypothesis H in Table II is actually a spurious hypothesis as shown in Table IV. The original hypothesis H indicates that treatment A is more effective than B. However, after we further divide the two subpopulations of H into smaller subgroups using attribute "stage," we find that treatment B is more effective than treatment A for patients at stage 1 and patients at stage 2. This phenomenon is called *Simpson's paradox* [Simpson 1951]. It is caused because "stage" has associations with both treatment and response: doctors tend to give treatment A to patients at stage 1 and treatment B to patients at stage 2; patients at stage 1 are easier to cure than patients at stage 2. Attributes that have associations with both the comparing attribute and the target attribute are called *confounding factors*.

2.5.1. Looking Deeper. The preceding examples show that further investigation of hypotheses is often very useful, and it can be conducted by dividing the two subpopulations under comparison into finer subgroups and then inspecting whether unexpected result is observed in pairs of the finer subgroups.

To divide the two subpopulations of a hypothesis H into smaller subgroups, we can add more items into the context P of H. A simple way to measure the impact of an item $A = v$ to H is to see how much the difference is lifted.

Definition 2.6 (*DiffLift(A=v|H), A=v is not in H*). Let $H = \langle P, A_{diff} = v_1|v_2, A_{target}, v_{target}\rangle$ be a hypothesis, $P_1 = P \cup \{A_{diff} = v_1\}$ and $P_2 = P \cup \{A_{diff} = v_2\}$ be the two subpopulations of H, and $A = v$ be an item not in H—that is, $A \neq A_{diff}$, $A \neq A_{target}$ and $A \notin P$. After adding item $A = v$ to P, we get two new subpopulations: $P_1' = P_1 \cup \{A = v\}$ and $P_2' = P_2 \cup \{A = v\}$. The lift of difference after adding $A = v$ to H is defined as $DiffLift(A{=}v|H){=}\frac{p_1'-p_2'}{p_1-p_2}$, where p_i is the proportion of v_{target} in subpopulation P_i, $p_1 \neq p_2$ and p_i' is the proportion of v_{target} in P_i', $i = 1, 2$.

We can divide the values of *DiffLift(A=v|H)* into three ranges, and each range represents a different situation.

Situation 1: *DiffLift(A=v|H)*>1. The new difference is wider than the old difference, and it is also of the same direction as the old difference. The larger the *DiffLift*, the more interesting the item. For example, in Table III, the *DiffLift*s of "time-of-failure=loading," "time-of-failure=in-operation," and "time-of-call=outputting" are 2.05, 0, and 0.05, respectively. Hence, "time-of-failure=loading" is a very interesting item.

Situation 2: $0 \leq$ *DiffLift(A=v|H)*≤ 1. The new difference is of the same direction as the old difference, but it is narrower than the old difference. The items satisfying this condition usually are not very interesting (e.g., "time-of-failure=in-operation" and "time-of-failure=outputting" in Table III).

Situation 3: *DiffLift(A=v|H)*< 0. The new difference is of the opposite direction of the old one. If the values of an attribute all satisfy this condition, then we have a Simpson's paradox as in Table IV, where *DiffLift(stage=1|H₁)*$= -0.683$ and *DiffLift(stage=2|H₁)* $= -0.833$.

Definition 2.7 (*Simpson's Paradox*). Given a hypothesis H and an attribute A not in H, if for every value v of A we have *DiffLift(A = v|H)*< 0, then we say that H and A form a Simpson's paradox.

In some cases, *DiffLift(A = v|H)* is not sufficient to capture all information. Let n_i be the size of subpopulation P_i and n_i' be the size of subpopulation P_i', $i = 1, 2$. If n_i' is extremely small compared with n_i, $i = 1, 2$, then even if *DiffLift(A = v|H)* is positive and very large, item $A = v$ can hardly have any material impact on H. We are more interested in finding those attribute values that have a big positive *DiffLift(A=v|H)* and are also associated with a large number of records. Acting upon such attribute values, we are able to make a much bigger impact than acting upon attribute values that are associated with few records.

If we divide subpopulation P_1 into several disjoint subsets $P_{11}, P_{12}, \ldots, P_{1k}$, then the proportion of v_{target} in P_1 can be expressed as $p_1 = \sum_{i=1}^{k} \frac{n_{1i}}{n_1} p_{1i}$, where n_1 is the size of subpopulation P_1, n_{1i} is the size of subset P_{1i}, and p_{1i} is the proportion of v_{target} in subset P_{1i}. Hence, $\frac{n_{1i}}{n_1}(p_{1i} - p_1)$ can be regarded as the overall contribution of P_{1i} to P_1, denoted as $Contribution(P_{1i}|P_1)$. We have $\sum_{i=1}^{k} \frac{n_{1i}}{n_1}(p_{1i} - p_1) = \sum_{i=1}^{k} \frac{n_{1i}}{n_1} p_{1i} - p_1 \frac{\sum_{i=1}^{k} n_{1i}}{n_1} = 0$. The contribution of $A = v$ to H is determined by which one is bigger, the contribution of P_1' to P_1, or the contribution of P_2' to P_2.

Definition 2.8 (*Contribution(A=v|H)*). Let H, $A = v$, P_i, p_i, P'_i, and p'_i, $i = 1, 2$, be defined as in Definition 2.6. The contribution of $A = v$ to H is defined as *Contribution(A=v|H)*=$(\frac{n'_1}{n_1}(p'_1 - p_1) - \frac{n'_2}{n_2}(p'_2 - p_2))/(p_1 - p_2)$.

If *Contribution(A* $= v|H) > 0$, we say that $A = v$ contributes positively to H; if *Contribution(A* $= v|H) < 0$, we say that $A = v$ contributes negatively to H; otherwise, we say that $A = v$ makes no contributions to H.

2.5.2. Looking Broader. The preceding analysis inspects a hypothesis in more specific contexts and studies the impact of the items outside of the hypothesis on the difference captured by the hypothesis. Another way of analyzing a hypothesis is to examine it in broader contexts and study the impact of the items within the hypothesis.

Definition 2.9 (*DiffLift(H, X), X is a subpattern of the context pattern of H*). Let $H = \langle P, A_{diff} = v_1|v_2, A_{target}, v_{target}\rangle$ be a hypothesis, $P_1 = P \cup \{A_{diff} = v_1\}$ and $P_2 = P \cup \{A_{diff} = v_2\}$ be the two subpopulations of H, X be a subpattern of P. After removing X from P, we get two new subpopulations: $P'_1 = P_1 - X$ and $P'_2 = P_2 - X$. The impact of X to H is defined as *DiffLift(H, X)*=$\frac{p_1-p_2}{p'_1-p'_2}$, where p_i is the proportion of v_{target} in subpopulation P_i and p'_i is the proportion of v_{target} in P'_i, $i = 1, 2$.

If *DiffLift(H, X)* $= 1$, then X has no impact on the difference captured by H. If *DiffLift(H, X)* is far from 1, then X is very important in H. If we set X to P, then we can see whether the difference captured by H still holds in the general population. If X contains only one item or contains exactly one less item than P, then we can get the impact of individual items.

2.6. Generalizations

Generalization to the continuous target attribute. When A_{target} is continuous, we can simply use m_i to replace p_i and m'_i to replace p'_i in Definitions 2.2, 2.3, 2.6, 2.8, and 2.9, where m_i and m'_i is the mean of A_{target} in subpopulation P_i and P'_i, respectively, $i = 1, 2$. If A_{target} is normally distributed, we can use the t-test [Gosset 1908] to calculate p-values; otherwise, we can use the Mann-Whitney test [Mann and Whitney 1947].

Generalization to hypotheses involving only one sample or more than two samples. We can simply represent a hypothesis involving one sample as $H = \langle P, A_{target}, v_{target}\rangle$. In this case, the statistics of subpopulation P is compared to the whole population. *DiffLift* and *Contribution* can be defined accordingly. It is also straightforward to formulate and test hypotheses involving more than two samples. A hypothesis involving k ($k > 2$) samples can be represented as $H = \langle P, A_{diff} = v_1|v_2| \cdots |v_k, A_{target}, v_{target}\rangle$. Statistical tests for more than two samples are available as well. However, the analysis of hypothesis involving more than two samples is much more complicated. We can resort to pairwise comparison and analysis.

2.7. Problem Statement

If the size of the two subpopulations under comparison is too small, statistical tests usually do not have enough power to detect the difference even if the difference is real. Hence, testing hypotheses involving very small subpopulations is often futile. To save computation cost, we put a minimum support constraint *min_sup* on the size of the two subpopulations. The minimum support threshold can be set based on the required false-positive rate and power [Witte et al. 2000].

Our system requires users to supply the following parameters: (1) a minimum support threshold *min_sup*; (2) a maximum p-value threshold *max_pvalue*, which indicates the level of statistical significance if one single hypothesis is tested; (3) a minimum difference threshold *min_diff*, which reflects the level of domain significance; (4) a

Table V. Example Association Rules

	Left-Hand Side Pattern	Right-Hand Side	Support	Confidence	Lift
R_1	Treatment=A, Stage=1	response=positive	880	90.9%	1.06
R_2	Treatment=B, Stage=1	response=positive	200	95%	1.10
R_3	Treatment=A, Stage=2	response=positive	120	75%	0.87
R_4	Treatment=B, Stage=2	response=positive	800	80%	0.93
R_5	Treatment=A	response=positive	1,000	89%	1.03
R_6	Treatment=B	response=positive	1,000	83%	0.97
R_7	Stage=1	response=positive	1,080	91.7%	1.07
R_8	Stage=2	response=positive	920	79.3%	0.92

maximum error threshold ϵ for removing redundant hypotheses; (5) a target attribute A_{target} and a target attribute value v_{target} if A_{target} is categorical; and (6) a set of grouping attributes $\mathcal{A}_{grouping}$, and the grouping attributes must be categorical. Users can set the parameters based on their requirements and domain knowledge. The last parameter is optional. If users do not specify the grouping attributes, then we use all categorical attributes in the given dataset as grouping attributes.

Given a dataset D and the preceding parameters, the hypothesis testing task aims to find a minimal set of hypotheses $\mathcal{H}_{rep}$ such that

—every $H = \langle P, A_{diff} = v_1|v_2, A_{target}, v_{target}\rangle \in \mathcal{H}_{rep}$ satisfies the following conditions:
(1) $A_{diff} \in \mathcal{A}_{grouping}$, and $\forall$ item $A = v$ in P, $A \in \mathcal{A}_{grouping}$.
(2) Context pattern P is a closed pattern.
(3) $sup(P_1) \geq min\_sup, sup(P_2) \geq min\_sup$, where $P_1 = P \cup \{A_{diff} = v_1\}$, $P_2 = P \cup \{A_{diff} = v_2\}$.
(4) p-value$(H) \leq max_pvalue$.
(5) If A_{target} is categorical, then $|p_1 - p_2| \geq min\_diff$, where p_i is the proportion of v_{target} in subpopulation P_i, $i = 1, 2$. If A_{target} is continuous, then $|m_1 - m_2| \geq min\_diff$, where m_i is the mean of A_{target} in subpopulation P_i, $i = 1, 2$.

—for every H' that satisfies the preceding conditions but H' is not included in $\mathcal{H}_{rep}$, there exists $H \in \mathcal{H}_{rep}$ such that H ϵ-covers H' (completeness).

—$\forall H \in \mathcal{H}_{rep}$, if H is removed from $\mathcal{H}_{rep}$, then there exists at least one significant hypothesis H' such that none of the remaining hypotheses in $\mathcal{H}_{rep}$ ϵ-covers H' (minimality).

2.8. Relation with Association Rule Mining

Association rule mining is a well-established problem in the data mining area. An association rule is of the form $X \Rightarrow Y$, where X and Y are two nonempty patterns. It means that if the left-hand side X occurs, then the right-hand side Y is also very likely to occur. Class association rule mining [Liu et al. 1998], contrast pattern mining [Bay and Pazzani 1999; Webb et al. 2003], emerging pattern mining [Dong and Li 1999], and subgroup discovery [Wrobel 1997] are special cases of association rule mining where the right-hand sides of rules are restricted to class labels or group labels [Novak et al. 2009].

Hypotheses and association rules are different forms of knowledge representation. Association rules focus on the collective effect of the items in the left-hand side pattern, whereas hypotheses study the effect of individual items (the comparing items) under certain conditions (the context). To form hypotheses, we need to put two or more rules together. Table V shows several example association rules, and the “support” column shows the support of the left-hand side patterns. The first rule R_1 shows that 90.9% of the patients at stage 1 respond positively to treatment A, which is higher than the overall positive response rate of 86% on the whole dataset. From this rule alone, it is

Table VI. Example Hypotheses

	Context	Comparing Attribute	Target Value	Support	Confidence	p-Value
H_1	{Stage=1}	Treatment=A	response=positive	880	90.9%	0.0805
	(1080, 91.7%)	Treatment=B		200	95%	
H_2	{Treatment=A}	Stage=1	response=positive	880	90.9%	3.99×10^{-7}
	(1000, 89%)	Stage=2		120	75%	

hard for users to conjecture which item contributes to the higher positive response rate: is it because treatment A is more effective than other treatments or because patients at stage 1 is easier to cure? Users need to look at other related rules, such as rule R_2 and R_3 in Table V, to have a clear picture. In this example, if we compare R_1 with its subrules R_5 and R_7, it is still not easy to have a clear idea because the confidence of both subrules is higher than the overall positive response rate of 86%. Association rule mining algorithms often produce a large number of rules and rank them using various interestingness measures. Related rules can be far apart from one another in the ranking, or even worse, some related rules are thrown away because they do not satisfy the interestingness criteria. For example, rules R_3, R_4, R_6, and R_8 may be thrown away because their confidence is lower than the overall positive response rate of 86%. However, these rules are useful for understanding other rules. It is time consuming for end users to manually search for related rules and do a comparative study of the rules that are interesting to them.

The work proposed in this article tackles the problem by putting related rules together to form tentative hypotheses. Table VI shows two example hypotheses formed from the rules in Table V. Hypothesis H_1 is formed from R_1 and R_2. It shows that treatment A is less effective than treatment B for patients at stage 1. Hypothesis H_2 is formed from R_1 and R_3, and it shows that patients at stage 1 are easier to cure than patients at stage 2 using treatment A. From these two hypotheses, users can easily see the impact of individual items in rule R_1.

Association rules and hypotheses allow users to look at data from different perspectives. We believe that they complement each other. Users can have a more comprehensive understanding of the data by exploring both types of knowledge representations. Another use of hypotheses is to find actionable knowledge. A significant hypothesis represents two rules that differ by only one item on the left-hand side and have very different confidence or other statistics with respect to the same right-hand side. In other words, it captures small changes that can make a big impact. It may suggest inexpensive actions that users can take to change things toward a desired direction.

3. AUTOMATIC HYPOTHESIS TESTING AND ANALYSIS

We generate hypotheses in two steps. In the first step, we generate large subpopulations using existing frequent pattern mining techniques. In the second step, we pair the large subpopulations up to form tentative hypotheses. Only representative significant hypotheses are retained.

3.1. Finding Large Subpopulations and Their Statistics

Generating large subpopulations containing no less than *min_sup* records is equivalent to mining frequent patterns with support no less than *min_sup*. Many efficient algorithms have been developed for mining frequent patterns [Goethals and Zaki 2003]. They can be used to generate large subpopulations. Existing frequent pattern mining algorithms collect only the support of frequent patterns. Additional information is needed for calculating p-values and for analyzing significant hypotheses. We modify the frequent pattern mining algorithm used in our system as follows.

We collect more statistics of the large subpopulations besides the support of the patterns defining them for the calculation of p-values. We take the two commonly used tests, the χ^2-test and t-test, as examples. Let P be a frequent pattern. When the χ^2-test is used, we need to collect $sup(P)$ and $sup(P \cup \{A_{target} = v_{target}\})$ to get the proportion of $A_{target} = v_{target}$ in subpopulation $T(P)$. When the t-test is used, we need to get the mean m_P and the standard deviation s_P of A_{target} in subpopulation $T(P)$. These two statistics are defined as follows: $m_P = \frac{\sum_{i=1}^{sup(P)} v_i}{sup(P)}$ and $s_P = \sqrt{\frac{\sum_{i=1}^{sup(P)}(v_i - m_P)^2}{sup(P)-1}} = \sqrt{\frac{\sum_{i=1}^{sup(P)} v_i^2 - sup(P) \cdot m_P^2}{sup(P)-1}}$, where v_i is the A_{target} value of the i-th record in subpopulation $T(P)$. Hence, we need to collect $sup(P)$, $\sum_{i=1}^{sup(P)} v_i$ and $\sum_{i=1}^{sup(P)} v_i^2$ to calculate m_P and s_P. All the preceding information can be collected as we count the support of pattern P. No additional scan of data is needed.

When looking deeper into a significant hypothesis H, we divide the two subpopulations of H into finer subpopulations by adding an item to the defining patterns of the two subpopulations. The resultant patterns may not be frequent. Hence, we need to generate not only frequent patterns but also some infrequent patterns to make sure that all information needed for analyzing a hypothesis is available. The generated infrequent patterns are the immediate superpatterns of some frequent patterns. In a regular frequent pattern mining algorithm, a pattern is not extended if it is not frequent. In the modified version, an infrequent pattern is still extended if it has at least one frequent immediate subpattern. To enable the checking of this condition during the mining process, we explore the search space in a way such that the subpatterns of a pattern are generated before the pattern itself. A pattern is not extended only if all of its immediate subpatterns are infrequent.

During the mining process, we check whether a frequent pattern is closed and mark those that are closed. Only frequent closed patterns are used as context patterns in the hypothesis generation step. In hypothesis generation and analysis, we need to retrieve a pattern's immediate superpatterns and subpatterns frequently as described in the next two subsections. When the number of patterns generated is very large, the cost for retrieving immediate superpatterns/subpatterns can be very high. Index structures for set-valued data, such as inverted files and signature files, can be used here to index frequent patterns. In our implementation, we use the CFP-tree structure [Liu et al. 2007], which is a compact structure specially designed for storing frequent patterns. It can be directly constructed by frequent pattern mining algorithms. CFP-tree supports efficient exact match and (immediate) subset/superset search. More details of CFP-tree can be found in Liu et al. [2007, 2013].

3.2. Generating Representative Hypotheses

The pseudocodes for generating representative hypotheses are shown in Algorithm 1. Given a hypothesis $H = \langle P, A_{diff} = v_1|v_2, A_{target}, v_{target}\rangle$, the defining patterns of the two subpopulations of H, $P_1 = P \cup \{A_{diff} = v_1\}$ and $P_2 = P \cup \{A_{diff} = v_2\}$, contain one more item than the context pattern P, so they are immediate superpatterns of P. The support of both P_1 and P_2 is no less than min_sup, and their subpopulations do not overlap. Therefore, the support of the context pattern is at least $2 \cdot min\_sup$. We generate representative hypotheses as follows. $\mathcal{H}_{rep}$ is used to store representative hypotheses that have been generated so far, and it is empty initially (line 1). For each frequent closed pattern P with support no less than $2 \cdot min\_sup$, we use it as a context and retrieve all of its frequent immediate superpatterns (line 3). We then group these superpatterns of P based on the attributes that do not appear in P (lines 4 and 5). Patterns that have the same attribute outside P are placed in the same group. Patterns in the same group are then paired up to form hypotheses (line 7), and the attribute outside P becomes

ALGORITHM 1: RepHyps

```
Input:
    $\mathcal{F}$: the set of frequent patterns;
    $A_{target}$: the target attribute;
    $v_{target}$: the target attribute value;
    $min\_sup$: the minimum support threshold;
    $max\_pvalue$: the maximum $p$-value threshold;
    $min\_diff$: the minimum difference threshold;
    $\epsilon$: the maximum error threshold for removing redundant hypotheses;
Output:
    $\mathcal{H}_{rep}$: a minimal set of representative significant hypotheses;
Description:
 1: $\mathcal{H}_{rep} = \{\}$;
 2: for all frequent closed pattern $P$ in $\mathcal{F}$ with support $\geq 2 \cdot min\_sup$ (the patterns are
    processed in a way such that all subsets of $P$ are processed before $P$) do
 3:   Retrieve all immediate superpatterns of $P$ with support $\geq min\_sup$, denoted as $\mathcal{S}$;
 4:   Let $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$ be the set of attributes in $\mathcal{S}$ but not in $P$.
 5:   Group patterns in $\mathcal{S}$ into $k$ groups based on the attributes in $\mathcal{A}$:
      $G_i = \{P' | P' \in \mathcal{S}, P'\ contains\ A_i\}, i = 1, 2, \ldots, k$;
 6:   for all $i = 1$ to $k$ do
 7:     for all pair of patterns $P_1$, $P_2$ in $G_i$ do
 8:       if the difference between subpopulations $P_1$ and $P_2 \geq min\_diff$ then
 9:         Calculate $p$-value of $H = \{P, A_i = v_1|v_2, A_{target}, v_{target}\}$, $A_i = v_j \in P_j$, $j = 1, 2$;
10:         if $p$-value$(H) \leq max\_pvalue$ and none of hypothesis in $\mathcal{H}_{rep}$ $\epsilon$-covers $H$ then
11:           $\mathcal{H}_{rep} = \mathcal{H}_{rep} \cup \{H\}$;
```

the comparing attribute (line 9). The context patterns are processed in the order such that the subpatterns of a pattern P are processed before pattern P (line 2). If a newly formed hypothesis H is statistically significant and it is not covered by any of the representative hypotheses that have been generated so far, then H is regarded as a representative hypothesis and is added to $\mathcal{H}_{rep}$ (lines 10 and 11).

LEMMA 3.1. *The set of representative hypotheses generated by Algorithm 1 covers all significant hypotheses satisfying the given constraints and is minimal.*

PROOF. A significant hypothesis H is discarded only when there already exists $H' \in \mathcal{H}_{rep}$ that covers H (line 10). Hence, every significant hypothesis is covered. A hypothesis H is added to $\mathcal{H}_{rep}$ only if none of the hypotheses that are already in $\mathcal{H}_{rep}$ covers it (lines 10 and 11). Since context patterns are processed in the order such that the subpatterns of a pattern P are processed before P (line 2), none of the hypotheses generated after H can cover H based on Lemma 2.5. Hence, H can only be covered by itself among all hypotheses in $\mathcal{H}_{rep}$. Therefore, $\mathcal{H}_{rep}$ is minimal. □

In Algorithm 1, we need to check whether a newly generated significant hypothesis H can be covered by some hypothesis in $\mathcal{H}_{rep}$ (line 10). For a hypothesis H' to cover H, it must have the same comparing items as H, and the context pattern of H' must be a subpattern of the context pattern of H. We use two methods to do the checking. One method divides representative hypotheses based on comparing items, and we denote this method as "RepHyps-compitems." More specifically, for each pair of comparing items $A_{diff} = v_1|v_2$, we maintain the set of representative hypotheses that have $A_{diff} = v_1|v_2$ as comparing items in a prefix-tree T_{A_{diff}, v_1, v_2}. The prefix-tree is built on the context patterns of the hypotheses. For each newly generated hypothesis $H = \{P, A_i = v_1|v_2, A_{target}, v_{target}\}$, we first locate prefix-tree T_{A_i, v_1, v_2} using the comparing items of H in a lookup table and then search for the subpatterns of P in the tree. For every subpattern that is found in the tree, we check whether the corresponding hypothesis covers H. If no hypothesis in T_{A_i,v_1,v_2} covers H, then H is regarded as a representative hypothesis and is inserted into the prefix-tree T_{A_i,v_1,v_2}; otherwise, H is regarded as redundant and is discarded.

The second method divides hypotheses based on context patterns, and we denote this method as "RepHyps-context." More specifically, all representative patterns in H_{rep} are stored in a prefix-tree T built on the context patterns of the hypotheses. For each context pattern P, we maintain the set of representative patterns that have P as context in a list. The hypotheses in the list are sorted based on the comparing items. For each frequent closed pattern P, we first generate all significant hypotheses that have P as context and store them in a list. We then search for the subpatterns of P in T. For each subpattern P' of P, we compare the hypothesis list of P with that of P'. Those hypotheses of P that can be covered by that of P' are removed from the list. If the hypothesis list of P becomes empty, then we stop the pruning. Otherwise, we continue to search for the subpatterns of P in T until all subpatterns of P are considered. The remaining hypotheses of P are representative hypotheses, and they are added to H_{rep}.

The preceding two pruning methods have some advantages and disadvantages. A context pattern P often forms significant hypotheses with multiple pairs of comparing items. The first method searches for the subpatterns of P in multiple small prefix-trees—one for each comparing item pair. This can be costly. The second method searches for the subpatterns of P in a large prefix-tree only once. However, it needs to sort and compare the hypothesis list of P with that of its subpatterns. This cost is not needed in the first method, as the first method uses a lookup table to locate comparing item pairs. Our experiment results show that both pruning methods are very efficient. They incur little overhead compared to the overall cost for hypothesis generation.

3.3. Generating Information for Hypothesis Analysis

Users may want to take a deeper or a broader look of a significant hypothesis to have a better understanding. If users want to look deeper into a hypothesis H, we generate the following information: (1) the set of Simpson's paradoxes formed by H with attributes not in H and (2) the list of items not in H ranked in descending order of $\mathit{DiffLift}(A = v|H)$ and $\mathit{Contribution}(A = v|H)$, respectively. The pseudocodes for generating the preceding information are shown in Algorithm 2.

ALGORITHM 2: LookDeeper

```
Input:
    Hypothesis $H = \{P, A_{diff} = v_1|v_2, A_{target}, v_{target}\}$;
    $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$ is the set of attributes not in $H$;
Description:
Retrieve the immediate superpatterns of $P_1 = P \cup \{A_{diff} = v_1\}$, denoted as $\mathcal{S}_1$;
Retrieve the immediate superpatterns of $P_2 = P \cup \{A_{diff} = v_2\}$, denoted as $\mathcal{S}_2$;
Pair superpatterns of $P_1$ and $P_2$ and then group them into $k$ groups based on attributes in
    $\mathcal{A}$: $G_i = \{(P'_1, P'_2)|P'_1 \in \mathcal{S}_1, P'_2 \in \mathcal{S}_2, P'_1$ and $P'_2$ contain a same value of $A_i\}, i = 1, 2, \ldots, k$;
for all $i = 1$ to $k$ do
    for all element $(P'_1, P'_2)$ in $G_i$ do
        Let $v$ be the value of $A_i$ contained in $P'_1$ and $P'_2$;
        Calculate $\mathit{DiffLift}(A_i = v|H)$ and $\mathit{Contribution}(A_i = v|H)$ using the statistics of $P_1$, $P_2$,
            $P'_1$, and $P'_2$;
    if $\forall v \in \mathit{Dom}(A_i), \mathit{DiffLift}(A_i = v|H) < 0$ then
        Output Simpson's paradox $(H, A_i)$;
Rank the items in descending order of DiffLift and Contribution, respectively;
```

Table VII. Datasets

Datasets	#instances	#attributes	A_{target}	v_{target}
Adult	48,842	15	class	>50K
German Credit	1,000	21	class	bad
mushroom	8,124	23	class	p
Thyroid	3,772	30	class	sick

Let P_1 and P_2 be the two subpopulations of H. We retrieve the immediate superpatterns of P_1 and P_2 (line 1 and 2). Let P_1' and P_2' be an immediate superpattern of P_1 and P_2, respectively. If P_1' and P_2' contain the same item that appears in neither P_1 nor P_2, P_1' and P_2' form a pair (P_1', P_2'). We group these pairs based on the attributes not in H (line 3). For each item $A = v$ that is not in H, *DiffLift*$(A = v|H)$ and *Contribution*$(A = v|H)$ are calculated using the statistics of P_1, P_2, $P_1 \cup \{A = v\}$, and $P_2 \cup \{A = v\}$ (lines 4 through 7). Simpson's paradoxes are generated based on *DiffLift* (lines 8 and 9). After all information for analysis are generated, we then output lists of items ranked in descending order of *DiffLift* or *Contribution* (line 10). At line 8, *Dom*(A_i) is the set of values taken by attribute A_i.

If users want to take a broader look of hypothesis $H = \{P, A_{diff} = v_1|v_2, A_{target}, v_{target}\}$, we generate the following information: (1) hypothesis $\{\{\}, A_{diff} = v_1|v_2, A_{target}, v_{target}\}$ and *DiffLift*(H, P); (2) hypotheses $H = \{P - \{x\}, A_{diff} = v_1|v_2, A_{target}, v_{target}\}$ and *DiffLift*(H, x), where x is an item in P; and (3) hypotheses $H = \{\{x\}, A_{diff} = v_1|v_2, A_{target}, v_{target}\}$ and *DiffLift*$(H, P - \{x\})$, where x is an item in P. The preceding information can be generated by searching for the statistics of the corresponding subpatterns of P_1 and P_2, where $P_i = P \cup \{A_{diff} = v_i\}$, $i = 1, 2$.

3.4. Complexity

Hypotheses are generated from frequent patterns. A frequent pattern P is paired with those frequent patterns that have the same set of attributes as P and differ by only one attribute value from P. Hence, the number of frequent patterns that can be paired with P is upper bounded by $\sum_{A \in P}(|Dom(A)| - 1)$, where $Dom(A)$ is the set of distinct values taken by attribute A. This number is within a fixed range, and it is usually very small compared to the number of frequent patterns. Hence, the number of hypotheses to be tested is linear to the number of frequent patterns.

The number of patterns in a dataset is exponential to the number of attributes in the dataset. On high-dimensional datasets, the cost for mining frequent patterns may be high. The problem can be alleviated with the advances of frequent pattern mining techniques. Given limited computation resources and limited manpower for interpreting the discovered knowledge, we can also put a maximum length constraint on frequent patterns to make a trade-off between the cost required and the knowledge that can be discoveredas longer patterns are harder to understand and the information captured by longer patterns may be already covered by shorter patterns.

4. A PERFORMANCE STUDY

In this section, we study the efficiency of the proposed algorithms and demonstrate the usefulness of the generated hypotheses via one case study. We conducted the experiments on a PC with a 3.4Ghz Intel Core i7-2600 CPU and 8GB RAM. The operating system is a 64-bit Windows 7. Our algorithms were implemented in C++ and compiled using Visual Studio 2010.

We use several datasets from the UCI Machine Learning Repository [Bache and Lichman 2013] in our experiments. Table VII shows some statistics of these datasets. Continuous attributes are discretized using MLC++ (http://www.sgi.com/tech/mlc/db/).

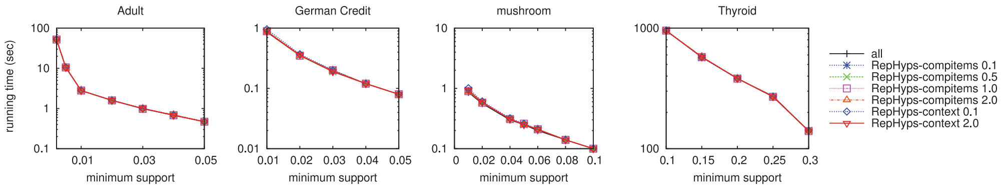

Fig. 1. Hypothesis generation time. On all datasets, $max\_pvalue = 0.05$, $min\_diff = 0$.

All of these datasets have a class attribute for classification. We use it as the target attribute and select one of its values as the target value.

4.1. Efficiency of Hypothesis Generation

This experiment studies the efficiency of our representative hypothesis generation algorithms. We use a variant of FP-growth [Han et al. 2000] to mine frequent patterns and store them in a CFP-tree [Liu et al. 2007], then generate hypotheses from the CFP-tree. The efficiency and scalability of the algorithms for frequent pattern mining has been studied extensively, so our focus here is on the hypothesis generation step.

Figure 1 shows the time for generating hypotheses from frequent patterns, and the time does not include the time for outputting hypotheses. In these figures, “all” denotes the algorithm that does not check the redundancy of hypotheses and outputs all significant hypotheses, “RepHyps-compitems” denotes the algorithm that groups representative hypotheses based on comparing items for pruning redundant hypotheses, and “RepHyps-context” denotes the algorithm that groups representative hypotheses based on context patterns for pruning. The numbers after the two algorithm names are the values of ϵ. The two representative hypothesis generation algorithms have very close performance, and they are not very sensitive to the value of ϵ. Compared to “all,” the two representative hypothesis generation algorithms require additional cost to eliminate redundant hypotheses. On all datasets used, the three algorithms have very similar performance. This indicates that checking the redundancy of hypotheses incur little overhead compared to the overall cost. The running time of the three algorithms increases with the decrease of *min_sup* because more patterns become frequent, and thus more hypotheses are tested.

Our algorithms need to access the original dataset only in the large subpopulation generation phase using frequent pattern mining techniques, so the size of the datasets that our algorithms can handle is the same as that of the state-of-the-art frequent pattern mining algorithms. The CFP-tree storing the frequent patterns may be large, but the cost for retrieving patterns from a CFP-tree is not sensitive to the size of the CFP-tree [Liu et al. 2013]. Hence, the running time of Algorithm 1 is not quadratic to the number of frequent patterns even though it needs to pair frequent patterns up for comparison. We observe that the running time of Algorithm 1 is usually nearly linear to the number of frequent patterns.

Figure 2 shows the number of representative hypotheses generated under different ϵ values. When $\epsilon = 0.1$, the number of representative patterns is already an order of magnitude smaller than the number of all significant hypotheses on datasets *German Credit* and *mushroom*, two orders of magnitude smaller on *Adult*, and several orders of magnitude smaller on dataset *Thyroid*. With the increase of ϵ, the number of representative hypotheses decreases.

4.2. Efficiency of Hypothesis Analysis

In this experiment, we study the average time needed to analyze a hypothesis. Table VIII shows the minimum support thresholds used, the size of the CFP-tree

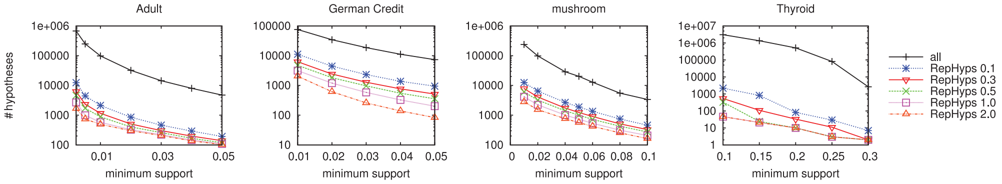

Fig. 2. Number of significant hypotheses. On all datasets, $max\_pvalue = 0.05$, $min\_diff = 0$.

Table VIII. Average Hypothesis Analysis Time

Datasets	*min_sup*	CFP-Tree Size (MB)	Hypotheses (#)	LookDeeper (ms)	LookBroader (ms)
Adult	0.005	92.89	4,453	4.3	0.45
Adult	0.05	7.87	193	3.2	0.21
German	0.01	29.60	11,194	2.1	0.30
German	0.05	7.01	946	2.1	0.17
mushroom	0.01	26.79	12,480	2.7	0.61
mushroom	0.1	2.94	460	2.9	0.63
Thyroid	0.1	423.42	2,177	2.39	0.91

Note: On all datasets, $\epsilon = 0.1$, $max\_pvalue = 0.05$ and $min\_diff = 0$.

constructed, the number of representative hypotheses generated, and the average analyzing time of the representative hypotheses. On all datasets, the average time for generating the information for looking deeper into one single hypothesis is below 5ms, and the average time for looking broader is below 1ms. These results suggest that it is possible to analyze hypotheses on demand in real time given that frequent patterns and their immediate infrequent superpatterns have been materialized and stored in a CFP-tree. Looking deeper takes longer than looking broader for two reasons. One reason is that the number of possible immediate superpatterns of a pattern is usually much larger than the number of its immediate subpatterns. The second reason is that CFP-tree is more efficient in supporting (immediate) subset search than (immediate) superset search as shown in Liu et al. [2013].

4.3. A Case Study: Dataset *Adult*

We use the dataset *Adult* as a case study to demonstrate the usefulness of the generated hypotheses in understanding the data and revealing new insights from the data. *Adult* contains the demographic information of 48,842 adults extracted from the 1994 U.S. Census database. It has been used in many studies to compare the performance of classification algorithms for predicting whether a person makes more than \$50K a year. In our study, we also choose the attribute *class* as the target attribute and *>50K* as the target value. On the whole dataset, 23.9% of adults make more than \$50K a year. We create a new attribute to replace attributes *capital-gain* and *capital-loss*. The value of the new attribute is the value of *capital-gain* minus the value of *capital-loss*. The new attribute is still named *capital-gain*.

Table IX shows several significant hypotheses observed in *Adult*. The first seven hypotheses are observed on the whole dataset, and their *p*-value is virtually 0. They indicate that *capital-gain*, *occupation*, *education*, *age*, *hours-per-week*, *marital-status*, and *sex* are associated with income. It is easy to understand that the first five attributes are important to income. It may not be that obvious why *marital-status* and *sex* are associated with income. We take a deeper look into these two hypotheses.

Table X shows the top-10 items with the highest contribution to hypothesis H_6 in Table IX. For both married and unmarried people, having a bachelor's degree increases their probability of earning more than \$50K (row 1). However, the increase

Table IX. Several Significant Hypotheses Identified on the Whole Dataset

ID	Context	Comparing Items	Sup	$P_{>50K}$	p-Value
H_1	{}	capital-gain=[7055.5, $+\infty$) capital-gain=[-731.5, 57)	2,055 42,560	98.6% 18.9%	0.000
H_2	{}	occupation=Exec-managerial occupation=Other-service	6,086 4,923	47.8% 4.14%	0.000
H_3	{}	education=Masters education=HS-grad	2,657 15,784	54.9% 15.9%	0.000
H_4	{}	age=[41.5, 54.5) age=[16, 21.5)	12,107 4,719	38.8% 0.21%	0.000
H_5	{}	hours-per-week=[49.5, 61.5) hours-per-week=(0, 34.5)	8,009 8,395	44.1% 6.91%	0.000
H_6	{}	marital-status=Married-civ-spouse marital-status=Never-married	22,379 16,117	44.6% 4.55%	0.000
H_7	{}	sex=Male sex=Female	32,650 16,192	30.4% 10.9%	0.000

Note: $P_{>50K}$ is a proportion of ">50K."

Table X. Looking Deeper into $H_6 = \langle \{\}$, *marital-status=Married-civ-spouse|Never-married*, class, >50K$\rangle$

					DiffLift	Contribution
	Item x	Comparing Items	Sup	$P_{>50K}$	$(x\|H_6)$	$(x\|H_6)$
1	education=Bachelors 41.3%	Married-civ-spouse Never-married	4,136 (18.5%) 2,681 (16.6%)	67.2% 9.8%	**1.43**	0.082
2	occupation=Exec-managerial 47.8%	Married-civ-spouse Never-married	**3,600 (16.1%)** **1,260 (7.82%)**	68.1% 13.6%	**1.36**	0.077
3	capital-gain=[7055.5, $+\infty$) 98.6%	Married-civ-spouse Never-married	**1,592 (7.11%)** **209 (1.30%)**	99.2% 97.1%	0.053	0.067
4	occupation=Prof-specialty 45.1%	Married-civ-spouse Never-married	3,182 (14.2%) 1,849 (11.5%)	70.8% 14.6%	**1.40**	0.064
5	age=[41.5, 54.5) 38.8%	Married-civ-spouse Never-married	**7,363 (32.9%)** **1,180 (7.32%)**	55.0% 16.4%	0.96	0.063
6	hours-per-week=[49.5, 61.5) 44.1%	Married-civ-spouse Never-married	**5,039 (22.5%)** **1,651 (10.2%)**	58.5% 15.6%	1.07	0.050
7	education=Masters 54.9%	Married-civ-spouse Never-married	1,527 (6.82%) 635 (3.94%)	76.5% 21.7%	**1.37**	0.037
8	workclass=Self-emp-inc 55.3%	Married-civ-spouse Never-married	**1,264 (5.65%)** **211 (1.31%)**	67.1% 14.2%	**1.32**	0.029
9	age=[16, 21.5) 0.21%	Married-civ-spouse Never-married	**163 (0.73%)** **4,473 (27.8%)**	3.1% 0.1%	0.074	0.023
10	education=Prof-school 74.0%	Married-civ-spouse Never-married	**596 (2.66%)** **139 (0.86%)**	84.6% 43.2%	1.03	0.018

Note: Numbers in column "Item x" are the proportion of >50K in the subpopulation defined by pattern $\{x\}$. The percentage numbers in column "Sup" are the percentage of records containing x in the two subpopulations of H_6, respectively.

for married people is more dramatic. As a result, a bachelor's degree enlarges the income difference between married and unmarried people by 1.43 times. Furthermore, a higher percentage of married people have a bachelor's degree than unmarried people. Therefore, item *education=Bachelors* has considerable contribution to the income difference between married and unmarried people. The same situation is also observed for items *occupation=Exec-managerial* (row 2), *occupation=Prof-specialty* (row 4), *education=Masters* (row 7), and *workclass=Self-emp-inc* (row 8). Having a capital

Table XI. Looking Deeper into $H_7 = \langle\{\}, \textit{sex=Male}|\textit{Female}, \text{class}, >50\text{K}\rangle$

		Comparing			DiffLift	Contribution
	Item x	Items	Sup	$P_{>50K}$	$(x\|H_7)$	$(x\|H_7)$
1	marital-status=Married-civ	Male	**19,899 (60.9%)**	44.6%	−0.011	0.179
	-spouse 44.6%	Female	**2,480 (15.3%)**	44.8%		
2	age=[41.5, 54.5)	Male	8,616 (26.4%)	47.2%	**1.51**	0.152
	38.8 %	Female	3,491 (21.6%)	17.9%		
3	hours-per-week=[49.5, 61.5)	Male	**6,696 (20.5%)**	47.4%	1.07	0.114
	44.1%	Female	**1,313 (8.1%)**	26.7%		
4	occupation=Exec-managerial	Male	4,338 (13.3%)	57.3%	**1.71**	0.111
	47.8%	Female	1,748 (10.8%)	24.1%		
5	education=Bachelors	Male	5,548 (17.0%)	50.3%	**1.51**	0.095
	41.3%	Female	2,477 (15.3%)	21.0%		
6	capital-gain=[7055.5, +∞)	Male	**1,701 (5.2%)**	99.1%	1.07	0.050
	98.6%	Female	**354 (2.2%)**	96.3%		
7	occupation=Sales	Male	3,557 (10.9%)	37.7%	**1.58**	0.066
	26.8%	Female	1,947 (12.0%)	6.9%		
8	workclass=Self-emp-inc	Male	**1484 (4.5%)**	58.9%	**1.47**	0.054
	55.3%	Female	**211 (1.3%)**	30.3%		
9	occupation=Prof-specialty	Male	3,930 (12.0%)	56.0%	**1.55**	0.052
	45.1%	Female	2,242 (13.8%)	26.0%		
10	age=[54.5, 61.5)	Male	2,630 (8.1%)	42.2%	**1.48**	0.041
	33.8%	Female	1,083 (6.7%)	13.4%		

Note: Numbers in column "Item x" are the proportion of >50K in the subpopulation defined by pattern $\{x\}$. The percentage numbers in column "Sup" are the percentage of records containing item x in the two subpopulations of H_7, respectively.

gain of at least $7,055.5 increases the probability of earning more than $50K a year from 0.239 to 0.986 (row 3). For married people, 7.11% of them have a capital gain of at least $7,055.5, whereas only 1.30% of unmarried people have a captital gain of at least $7,055.5. Hence, item *capital-gain=[7055.5, +∞)* is another reason for the income difference between married and unmarried people. Older people have more experiences and qualifications, so they usually earn more. About 38.8% of people between 41.5 and 54.5 years old earn more than $50K a year (row 5), whereas only 0.21% of people younger than 21.5 years earn more than $50K a year (row 9). Among the married people, 32.9% of them are between 41.5 and 54.5 years old, and only 0.73% of them are younger than 21.5 years. Among the unmarried people, 7.32% of them are between 41.5 and 54.5 years old, and 27.8% of them are younger than 21.5 years. Therefore, age is an important factor for the income difference between married and unmarried people. Working longer often means higher income. The average working hours per week on the whole dataset is 40.4. A higher percentage of married people are willing to work longer hours to support their families. More specifically, 22.5% of married people work between 49.5 and 61.5 hours per week, whereas only 10.2% of unmarried people work that long (row 6).

Table XI shows the top-10 items with the highest contribution to H_7 in Table IX. Hypothesis H_6 in Table IX shows that married people are more likely to earn more than $50K a year. In the dataset *Adult*, a much higher percentage of men are married than women (row 1), probably because many women quit their jobs to become housewives after marriage. For married people, women actually have a slightly higher chance to earn more than $50K than men. Earning more may be one of the reasons women stay at work after marriage. For experienced workers between 41.5 and 61.5 years old, the income difference between men and women is larger than that of the general

Table XII. Three Example Significant Hypotheses Identified on a Subset of Dataset *Adult*

ID	Context	Comparing Items	Sup	$P_{>50K}$	p-Value
H_8	{education=Masters}	occupation=Exec-managerial occupation=Prof-specialty	779 1,302	73.3% 48.7%	9.03E-029
H_9	{hours-per-week=(0, 34.5), marital-status=Married-civ-spouse}	sex=Male sex=Female	1,508 641	18.2% 36.5%	8.17E-019
H_{10}	{capital-gain=[−731.5, 57), Race=White}	Occupation=Craft-repair Occupation=Adm-clerical	4,821 4,128	19.2% 11.9%	8.84E-022

Table XIII. A Broader Look at Hypothesis $H_8 = \langle$ {education=Masters}, *occupation=Exec-managerial|Prof-specialty*, class, >50K$\rangle$

Context	Comparing Items	Sup	$P_{>50K}$	p-Value	Removed Item x	DiffLift(H_8, x)
{}	occupation=Exec-managerial occupation=Prof-specialty	6,086 6,172	47.8% 45.1%	0.0032	education=Masters	9.20

Table XIV. A Broader look at Hypothesis $H_9 = \langle$ {*hours-per-week=(0, 34.5), marital-status=Married-civ-spouse*}, *sex=Male|Female*, class, >50K$\rangle$

Context	Comparing Items	Sup	$P_{>50K}$	p-Value	Removed Item x	DiffLift(H_8, x)
{hours-per-week= (0, 34.5)}	sex=Male sex=Female	3,964 4,431	7.7% 6.2%	0.0058	marital-status= Married-civ-spouse	–11.9
{marital-status= Married-civ-spouse}	sex=Male sex=Female	19,899 2,480	44.6% 44.8%	0.847	hours-per-week= (0, 34.5)	87.75

population (rows 2 and 10). Men are more willing to work long hours than women (row 3): 20.5% of men work 49.5 to 61.5 hours per week, whereas only 8.1% of women work that long. For certain occupations, such as *Exec-managerial* (row 4), *Sales* (row 7), and *Prof-specialty* (row 9), the income difference between the two genders is larger than that of the general population. Among people with a bachelor's degree, the income difference between the two genders is also larger than that of the general population. Row 6 shows that a higher percentage of men have a capital gain of at least $7,055.5. Row 8 shows that a higher percentage of men start their own companies than women.

Table XII shows three significant hypotheses that are observed on a subset of the dataset. Hypothesis H_8 shows that for people with a master's degree, executive managers are much more likely to earn more than $50K than those doing professional/ specialty work. We take a broader look of this hypothesis on the whole dataset. On the whole dataset, the difference between the two occupations is relatively small, as shown in Table XIII. Hence, context pattern {*education=Masters*} is important to the difference between the two occupations depicted by H_8.

Hypothesis H_9 in Table XII shows that for married people working less than 34.5 hours per week, women are twice more likely to earn more than $50K a year than men, whereas in the whole population, men are about three times more likely to earn more than $50K a year than women, as indicated by H_7 in Table IX. A broader look at H_9 shows that for either married people or people working less than 34.5 hours per week, men and women have similar chance of earning more than $50K a year, as shown in Table XIV. Hence, both items in the context pattern of H_9 are important to the income difference between the two genders captured by H_9. This analysis shows that working shorter hours while earning more keeps married women in the workforce.

Hypothesis H_{10} in Table XII shows that among white people with little or no capital gain, craft repairers earn more than administration clerks. However, if we take a deeper look by dividing the two subpopulations using the attribute *sex*, we observe the opposite phenomenon in men and women. In other words, as shown in Table XV, for both genders, craft repairers actually earn less than administration clerks. This

Table XV. A Simpson's Paradox behind H_{10}

Context	Extra Attribute	Comparing Items	Sup	$P_{>50K}$
{capital-gain=[–731.5, 57), Race=White}	Sex = Male	Occupation=Craft-repair	4,599	19.9%
		Occupation=Adm-clerical	1,341	21.8%
	Sex = Female	Occupation=Craft-repair	222	5.9%
		Occupation=Adm-clerical	2,787	7.1%

Simpson's paradox is caused because men earn much more than women on average, and many more men work as craft repairers than women. Hence, the attribute *sex* is a confounding factor here. It has associations with both the comparing attribute and the target attribute. We need to be extra cautious when interpreting hypotheses involving a Simpson's paradox.

5. RELATED WORK

Exploratory hypothesis testing needs to explore a large space of tentative hypotheses. We employ association rule mining techniques for efficient exploration. Association rule mining was first proposed in Agrawal et al. [1993], and it has become an important problem in the data mining area since then. Many efficient algorithms have been proposed. A frequent itemset mining implementations repository has been set up [Goethals and Zaki 2003] (http://fimi.ua.ac.be/src/).

Association rule mining algorithms often produce a large number of rules, and not all of them are interesting. Various interestingness measures have been proposed to capture the interestingness of rules. Tan et al. [2002] and Geng and Hamilton [2006] surveyed various measures proposed in the literature. The statistical significance of rules has also been studied in some work [Webb 2007, 2008, 2010, 2011; Kirsch et al. 2009]. Many existing rule interestingness measures involve no comparison, and they look at one rule at a time. For interestingness measures that involve comparison, the comparison is usually between subpatterns and superpatterns. Exploratory hypothesis testing compares rules that differ by one item to find deviations. We also present the findings in the form of comparison (hypotheses), which allows users to look at the data from another perspective. It is often easier to comprehend when related information is presented together. Furthermore, we do not stop at simply comparing two rules. We also investigate the reasons behind the difference and look at issues such as Simpson' paradox. Some interestingness measures [Silberschatz and Tuzhilin 1995; Liu et al. 1999] compare rules with existing knowledge or expectations of users. The main difficulty faced by this approach is that there is no one general method that can represent all types of knowledge, and it is often very tedious and difficult for users to specify everything they know.

One work that is very related to ours is Freitas [1998]. This work compares the class label of a rule with that of its minimum generalizations (subpatterns) and uses the proportion of the minimum generalizations that have a different class label as a surprisingness measure. The outputs of the work are still rules. It is difficult for users to conjecture with respect to which minimum generalization the rule is surprising. In the same paper, Freitas also proposes an algorithm to identify Simpson's paradox based on the change of class labels. In our work, Simpson's paradox is one special output of the looking deeper operation. We provide several other methods for users to investigate the hypotheses that are interesting to them.

Liu et al. [2006] and Zhao et al. [2006] propose a system called *Opportunity Map* to support exploratory analysis of cellular phone call records, which casts rule analysis as OLAP operations and general impression mining. The same group of authors later found that although the operations on rule cubes are flexible, each operation is primitive and has to be initiated by the user. Finding a piece of actionable knowledge typically involves many operations and intense visual inspection. To solve this problem, they proposed the idea of identifying actionable knowledge via automated comparison [Zhang et al. 2009]. In their approach, users need to manually select two attribute values for comparison, and the system then ranks all other attributes based on their levels of interestingness. What they do is similar to the looking deeper operation in our approach, but the hypotheses in their system are provided by users manually instead of being generated automatically as in our approach. They do not identify Simpson's paradoxes either.

Another work that supports exploratory data analysis is Panda et al. [2010], which proposes the model summary problem. Model summaries are defined as partial dependence plots between a set of attributes and a variable predicted by a data mining model. These summaries are useful for understanding the impact of the attributes to the output of the data mining model. Efficient algorithms are proposed for tree-based models, and a user-friendly system called *Scolopax* is developed to support exploratory analysis of bird-sighting data [Okcan et al. 2013]. Since the algorithms are model specific, new algorithms are needed for other data mining models.

6. DISCUSSION AND CONCLUSION

In this article, we have formulated the exploratory hypothesis testing and analysis problem and proposed a data mining approach to solve the problem. Conventional hypothesis testing allows just one or a few hypotheses to be tested at one time, whereas exploratory hypothesis testing enables researchers to use computational methods to examine large numbers of hypotheses and to identify those that have a reasonable chance of being true. In this fashion, human oversights and limitations can be complemented by computers.

We used the absolute difference between two subpopulations to measure the domain significance. Other measures can be used here as well. For example, odds ratio and relative risk have been commonly used when the target attribute is nominal. For hypothesis analysis, we have defined two simple measures, *Contribution* and *DiffLift*, to measure the impact of an item to a hypothesis. We do not claim that the measures used in the article are the best, but we believe that they do capture useful information and can serve the purpose. In association rule mining, many interestingness measures have been proposed, but none of them is superior to all others in every aspect. The situation is the same here. It will be difficult to find a measure that is better than all other possible measures in every aspect. The choice of proper measures often depends on the need of the applications.

It is not our intention to replace conventional hypothesis testing with exploratory hypothesis testing. Instead, we believe that the two approaches are complementary to each other. Exploratory hypothesis testing can be employed to find potentially interesting things in the data via extensive computation, which is tedious and time consuming for scientists to do manually, especially with the large-scale datasets available today. The generated (representative) significant hypotheses provide starting points for scientists to examine further. To confirm these hypotheses, scientists still need to perform a rigorous evaluation using conventional hypothesis testing.

REFERENCES

Herve Abdi. 2007. Bonferroni and Šidák corrections for multiple comparisons. In *Encyclopedia of Measurements and Statistics*, N. J. Salkind (Ed.). Sage, Thousand Oaks, CA.

Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. 1993. Mining association rules between sets of items in large databases. In *Proceedings of SIGMOD*. 207–216.

Kevin Bache and Moshe Lichman. 2013. UCI Machine Learning Repository. Retrieved April 17, 2015, from http://archive.ics.uci.edu/ml.

Stephen D. Bay and Michael J. Pazzani. 1999. Detecting change in categorical data: Mining contrast sets. In *Proceedings of SIGKDD*. 302–306.

Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society* 57, 1, 125–133.

Guozhu Dong and Jinyan Li. 1999. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of SIGKDD*. 43–52.

Alex A. Freitas. 1998. On objective measures of rule surprisingness. In *Proceedings of PKDD*. 1–9.

Liqiang Geng and Howard J. Hamilton. 2006. Interestingness measures for data mining: A survey. *ACM Computing Surveys* 38, 3, Article No. 9.

Bart Goethals and Mohammed Javeed Zaki. 2003. FIMI'03: Workshop on frequent itemset mining implementations. In *Proceedings of ICDM*.

William Sealy Gosset. 1908. The probable error of a mean. *Biometrika* 6, 1, 1–25.

Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. In *Proceedings of SIGMOD*. 1–12.

Adam Kirsch, Michael Mitzenmacher, Andrea Pietracaprina, Geppino Pucci, Eli Upfal, and Fabio Vandin. 2009. An efficient rigorous approach for identifying statistically significant frequent itemsets. In *Proceedings of PODS*. 117–126.

Bing Liu, Wynne Hsu, and Yiming Ma. 1998. Integrating classification and association rule mining. In *Proceedings of SIGKDD*. 80–86.

Bing Liu, Wynne Hsu, Lai-Fun Mun, and Hing-Yan Lee. 1999. Finding interesting patterns using user expectations. *IEEE Transactions on Knowledge and Data Engineering* 11, 6, 817–832.

Bing Liu, Kaidi Zhao, Jeffrey Benkler, and Weimin Xiao. 2006. Rule interestingness analysis using OLAP operations. In *Proceedings of SIGKDD*. 297–306.

Guimei Liu, Mengling Feng, Yue Wang, Limsoon Wong, See-Kiong Ng, Tzia Liang Mah, and Edmund Jon Deoon Lee. 2011. Towards exploratory hypothesis testing and analysis. In *Proceedings of ICDE*. 745–756.

Guimei Liu, Hongjun Lu, and Jeffrey Xu Yu. 2007. CFP-tree: A compact disk-based structure for storing and querying frequent itemsets. *Information Systems* 32, 2, 295–319.

Guimei Liu, Andre Suchitra, and Limsoon Wong. 2013. A performance study of three disk-based structures for indexing and querying frequent itemsets. *PVLDB* 6, 7 (2013), 505–516.

Guimei Liu, Haojun Zhang, and Limsoon Wong. 2011. Controlling false positives in association rule mining. *Proceedings of the VLDB Endowment* 5, 2, 145–156.

Henry B. Mann and Donald R. Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics* 18, 50–60.

Harvey Motulsky. 1995. *Intuitive Biostatistics*. Oxford University Press.

Petra Kralj Novak, Nada Lavrac, and Geoffrey I. Webb. 2009. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research* 10, 377–403.

Alper Okcan, Mirek Riedewald, Biswanath Panda, and Daniel Fink. 2013. Scolopax: Exploratory analysis of scientific data. *Proceedings of the VLDB Endowment* 6, 12, 1298–1301.

Biswanath Panda, Mirek Riedewald, and Daniel Fink. 2010. The model-summary problem and a solution for trees. In *Proceedings of ICDE*. 449–460.

Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. 1999. Discovering frequent closed itemsets for association rules. In *Proceedings of ICDT*. 398–416.

Abraham Silberschatz and Alexander Tuzhilin. 1995. On subjective measures of interestingness in knowledge discovery. In *Proceedings of SIGKDD*. 275–281.

Edward H. Simpson. 1951. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society* 13, 2, 238–241.

Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. 2002. Selecting the right interestingness measure for association patterns. In *Proceedings of SIGKDD*. 32–41.

Geoffrey I. Webb. 2007. Discovering significant patterns. *Machine Learning* 68, 1, 1–33.

Geoffrey I. Webb. 2008. Layered critical values: A powerful direct-adjustment approach to discovering significant patterns. *Machine Learning* 71, 2–3, 307–323.

Geoffrey I. Webb. 2010. Self-sufficient itemsets: An approach to screening potentially interesting associations between items. *ACM Transactions on Knowledge Discovery from Data* 4, 1, Article No. 3.

Geoffrey I. Webb. 2011. Filtered-top-k association discovery. *Data Mining and Knowledge Discovery* 1, 3, 183–192.

Geoffrey I. Webb, Shane M. Butler, and Douglas A. Newlands. 2003. On detecting differences between groups. In *Proceedings of SIGKDD*. 256–265.

John S. Witte, Robert C. Elston, and Lon R. Cardon. 2000. On the relative sample size required for multiple comparison. *Statistics in Medicine* 16, 369–372.

Stefan Wrobel. 1997. An algorithm for multi-relational discovery of subgroups. In *Proceedings of PKDD*. 78–87.

Frank Yates. 1934. Contingency table involving small numbers and the χ^2 test. *Journal of the Royal Statistical Society* 1, 2, 217–235.

Lei Zhang, Bing Liu, Jeffrey Benkler, and Chi Zhou. 2009. Finding actionable knowledge via automated comparison. In *Proceedings of ICDE*. 1419–1430.

Kaidi Zhao, Bing Liu, Jeffrey Benkler, and Weimin Xiao. 2006. Opportunity map: Identifying causes of failure—a deployed data mining system. In *Proceedings of SIGKDD*. 892–901.

Received January 2014; revised September 2014; accepted December 2014