Statistically Significant Pattern Mining With Ordinal Utility

Thien Q. Tran®, Kazuto Fukuchi®, Youhei Akimoto®, and Jun Sakuma®

Abstract—Statistically significant pattern mining (SSPM), which evaluates each pattern via a hypothesis test, is an essential and challenging data mining task for knowledge discovery. We introduce a preference relation between patterns and aim to discover the most preferred patterns under the constraint of statistical significance, which has never been considered in existing SSPM problems. We propose an iterative multiple testing procedure that can alternately reject a hypothesis and safely ignore the less useful hypotheses than the rejected one. By filtering out patterns with low utility, we can avoid the significance budget consumption of rejecting useless (uninteresting) patterns and focus the significance budget on more useful patterns, leading to more useful discoveries. We show that the proposed method can control the familywise error rate (FWER) under certain assumptions, which can be satisfied by a realistic problem class in SSPM. We also show that the proposed method always discovers equally or more useful patterns than Tarone-Bonferroni and Subfamily-wise Multiple Testing (SMT). Finally, we conducted several experiments with both synthetic and real-world data to evaluate the performance of our method. The proposed method discovered many more useful patterns in the experiments with real-world datasets than the existing method for all five conducted tasks.

Index Terms—High-utility pattern, multiple testing, significant pattern mining

1 INTRODUCTION

Statistically significant pattern mining (SSPM) is the task of finding patterns that statistically occur more often in the data for one class than for another. Different from other pattern mining frameworks, the discovery of each pattern in SSPM is evaluated via a statistical hypothesis test. SSPM aims to maximize the number of true discoveries, i.e., minimize type-II error while controlling the number of false discoveries, i.e., controlling type-I error. Since discovered patterns are provided with a statistical guarantee, SSPM is important and widely applied in genetics, healthcare, and market analysis. For example, discoveries of treatment combinations are significantly efficacious in healthcare and could improve healthcare service quality. Similarly, medical scientists are interested in finding significant patterns of gene alleles associated with the onset of disease.

This study focuses on the utility of patterns and introduces a novel utility-oriented setting for SSPM. Our goal is to discover the most desired significant patterns regarding some utility criterion. The user can define such a utility criterion based on prior knowledge or preference of the patterns’ importance or interestingness. Considering the utility of patterns is essential in many applications spanning medicine, finance, and e-commerce, and many methods have been proposed [1]. To the best of our knowledge, this work is the first that considers the utility of pattern utility in the SSPM setting. Moreover, despite many off-the-shelf solutions for utility-oriented pattern mining (UPM), these methods do not assess the significance of the discovered pattern. They thus can not fulfill the statistical guarantee in the SSPM setting.

We first give an example to illustrate our proposed problem. Assume that we have the Adult dataset in which each transaction contains several demographic attribute values as explanatory variables and a binary target class (if income > 50K or not). In general, SSPM methods aim to find the patterns of demographic attributes (e.g., “university graduate” and “works 60 hours/week”) significantly associated with “income > 50K”. Herein, we introduce a preference relation between patterns and aim to find the most useful patterns under the constraint of statistical significance. Continuing with the same example, assume that we are more interested in patterns that realize “higher income with less education” than “higher income with higher education”; we can define patterns with “less education” as more preferred than patterns with “higher education”.

This paper proposes SPUR (Significant Pattern mining with Utility Relationship) to discover statistically significant patterns with the highest utility. Three main challenges must be solved to achieve this goal.

First, our goal is different from the common goal of SSPM. Generally, SSPM aims to discover as many patterns as possible while controlling the type-I error. On the other hand, our goal is to discover patterns with the highest utility while controlling the type-I error. Therefore, existing SSPM
methods, such as Tarone-Bonferroni (T-Bonferroni), are not necessarily practical for our setting. Since these methods do not consider the preference between patterns, they possibly waste the significance budget for controlling the type-I error of patterns that are not useful. Here, we propose an iterative discovering framework that ignores all the patterns that are less useful than the discovered pattern after discovering a pattern. Our method can save the significance budget for more useful patterns by ignoring less useful ones, which is expected to result in more useful discoveries without violating the type-I error constraint.

Fig. 1 demonstrates the statistically significant patterns in the Adult dataset found by T-Bonferroni (top) and our method SPUR (bottom). Specifically, we show the discovered patterns of “education level” and “work hour” while fixing other demographic features. We defined “less education” and “less hours-per-week” as more preferred than “higher education” and “more hours-per-week”. As we see from the figure, in two different demographic conditions, SPUR can find more preferred (i.e., “less education and less hours-per-week” but “higher-income”) patterns than the existing method with fewer rejections.

Second, since each discovery in SSPM is evaluated via a statistical test, the proposed approach is only applicable if we can develop an iterative multiple-testing procedure that can safely ignore less useful hypotheses after rejecting a hypothesis. By “safely”, we mean it does not violate the type-I error constraint. To our knowledge, no existing multiple-testing procedure can fulfill this requirement. We discovered that this is achievable for a specific SSPM problem class. We propose a method that can control the FWER under a particular assumption about the independence of p-values, in that the p-values of false null and true null hypotheses are mutually independent. In the SSPM context, this assumption can be satisfied when Fisher’s exact test is used, and the setup of patterns guarantees no overlap of samples between distinct patterns (see Section 3.3 for more details). We note that such a setting fits most of our intended applications.

Finally, because in a typical SSPM setting, we often have to consider many patterns, the number of hypotheses in the multiple-testing problem is also large. When the number of tests is large, simple multiple-testing adjustments, such as the Bonferroni correction, will be too conservative to discover significant patterns. Recently, many studies focus on the Tarone’s trick [2], which takes account of “untestable” hypotheses, resulting in more discoveries while still controlling the FWER. Thus, to be practical in the SSPM setting, we must also be able to leverage Tarone’s trick.

Our contributions in this work are as follows:

- We introduce the problem of discovering statistically significant patterns with the highest utility, given the ordinal utility between patterns. Then, we propose an iterative multiple-testing method that can reject more useful hypotheses by carefully ignoring less useful hypotheses at each iteration.
- We prove that SPUR controls the FWER under a particular assumption (independence of p-values between true and false hypotheses), which can be satisfied by a realistic SSPM problem class (Theorem 6.1).
- We prove that SPUR always achieves an equal or higher utility than Tarone-Bonferroni (Theorem 6.2) and SMT (Theorem 6.2).
- We conduct synthetic and real-world experiments to show the benefit of SPUR in discovering useful patterns. Our results show that SPUR can discover more useful patterns in all five real-world tasks.

This manuscript extends the previous version [3] as follows. First, we newly consider two more sophisticated comparison methods SMT and i-FWER. We discuss how these methods could be applied to our setting and their limitations. Second, we significantly improve the theoretical analysis of the Type-I error and utility guarantees. Particularly, we provide the full proof for the FWER guarantee theorem, our most important contribution. For the utility guarantee, we newly prove that SPUR always outperforms SMT. Third, we introduced new synthetic experiments, which are not only for comparing two new comparison methods but also provide a much deeper understanding of the behavior of SPUR and other methods. Finally, we further discussed relation to existing high-utility pattern mining works (Section 2.2), how real-world data are prepared (Appendix C, available online, and how to obtain the adjusted p-values (Appendix D, available in the online supplemental material).

2 RELATED WORK

2.1 Statistically Significant Pattern Mining

The most fundamental challenge in SSPM is the explosion in the number of patterns because many items have to be considered. Some methods have been proposed to overcome this challenge in improving discovery power and navigating complexity. After the early works in [4] and [5], an efficient method that combines the Tarone’s trick[2] and the association rule mining algorithm Apriori[6], named Limitless Arity Multiple-testing Procedure (LAMP), was proposed in [7]. Further studies have attempted to improve or extend LAMP to other settings [8], [9].

A body of literature focuses on other types of tests or other aspects of the discovering task. For example, [10]...
studied the Westfall-Young permutation test to deal with the dependence between patterns. [11] worked on Cochran-Mantel-Haenszel and [12] focused on Barnard’s exact test. Moreover, [13] focused on the statistical emerging pattern mining problem, while [14] focused on finding significant interactions between continuous features. However, to our knowledge, no SSPM studies have previously focused on the utility of patterns.

### 2.2 High-Utility Pattern Mining

Utility-oriented pattern mining (UPM) is a well-studied problem that aims to consider both the association (frequency) and the benefit of the patterns. UPM is important in numerous applications because it can incorporate the user’s domain knowledge and preference. In the last decade, a large number of UPM methods have been proposed, focusing on many types of utility, including high-utility item-sets [15], high-utility association rules [16] and high-utility sequential rules [17]. Also, various approaches have been proposed, e.g., Apriori-based, tree-based, projection-based, and vertical-horizontal-data-based [1].

However, despite many off-the-shelf solutions, these methods do not assess the observed pattern’s statistical significance as required in our SSPM setting. Performing the statistical testing afterward with the selected patterns by UPM is also problematic due to the selection bias. For example, assume that there are 100 patterns in the dataset and none of them is associated with the target class. In other words, all of these hypotheses are true null hypotheses, and none of them should be rejected. If we randomly select ten patterns and perform the tests, we can safely control the Type-I error under the significance level \( \alpha \). However, if we select the most promising ten patterns using a UPM method and conduct the tests using the same data afterward, we will fail to control the Type-I error. That is because these hypotheses are tested using the data associated just by chance. Thus, these selected hypotheses will more likely to be rejected than they should be.

Our work has a discovery interest that differs from existing UPM works. In particular, most UPM works aim to discover the frequent itemsets or association rules whose utility is no less than a user-specified minimum utility threshold. On the other hand, we want to find the most useful patterns under the statistical constraint. Here, we do not assume anything about the minimum utility threshold. Our method works even when the user cannot define or does not have a specific minimum utility threshold. Furthermore, our method can also be used along with the minimum utility threshold. If a minimum utility threshold is available, one can first filter the patterns under the minimum utility threshold and use SPUR for the remaining patterns. Because the number of patterns is reduced, each hypothesis can consume a larger significant budget, and more patterns are expected to be discovered.

### 2.3 Multiple Testing Framework With Side Information

Our work also relates to the multiple testing frameworks that aim to improve the testing power by involving side information. Many of these works focus on controlling the false discovery rate and exploit a prior ordering to focus more significant budget on more “promising” hypotheses [18], [19], [20]. Here, “promising” means that hypothesis is likely be rejected based on some prior knowledge. Recently, Dual Boyan et al. proposed an interactive method named i-FWER to control the familywise error rate in a similar setting [21]. i-FWER uses the “masking and unmasking” strategy [20] that allows users to interact with the testing procedure with a complex form of prior information rather than an undifferentiated list.

We emphasize that our setting differs from these works, and it is not straightforward to adapt these approaches to achieve our goal. First, in our setting, the side information is not used to define how promising a hypothesis is but how that hypothesis is preferable to the user. Second, in these approaches, one can only reliably obtain the rejected set at the end of the procedure. Thus, at an iteration \( t \) because we do not know which pattern will be surely rejected, there is no reliable way to remove unpromising hypotheses (patterns). Using these uncertain temporary results to remove less useful hypotheses, one would accidentally ignore useful discoveries that should not be ignored.

### 2.4 Hypotheses Stream Testing

Webb et al. introduced the subfamily-wise multiple testing (SMT), which focuses on the hypotheses stream problem [22]. Given a stream of hypotheses set \( H_0, H_1, \ldots, H_n \), SMT aims to sequentially reject one hypothesis for each set while controlling the FWER. Our setting can be considered a special case of SMT in which the hypotheses set can only shrink, i.e., \( H_t \subset H_{t-1} \). Our method leveraged this characteristic to achieve a significantly better testing power than SMT. We also show that the proposed method is compatible with the Tarone-trick which is crucial in SSPM setting.

### 3 Preliminaries

#### 3.1 Statistical Association Testing

Statistical association testing is a procedure for testing whether two random variables are statistically dependent. In the context of pattern mining, this is a procedure to test whether the presence of a pattern associates with the occurrence of a specific event, represented by a class label. Suppose that we have a dataset \( D = \{t_1, t_2, \ldots, t_{n_D}\} \) that contains \( n_D \) transactions defined in the universe of \( m \) items \( I = \{1, \ldots, m\} \). Here, a transaction \( t_i \) can be described by a vector \( x_i \) of length \( m \) and a binary-class label \( y_i \). The vector \( x_i \) indicates whether the corresponding items present in the transaction; i.e., \( x_{ij} = 1 \) if item \( j \) appears in transaction \( t_i \). We call a set of items \( s \subseteq I \) a pattern and use \( s \leq t \) to denote that “pattern \( s \) is included in transaction \( t_i \)” i.e., \( \forall j \in s, x_{ij} = 1 \). Moreover, we define the indicator \( X_{i,s} = \prod_{j \in s} x_{ij} \) in which \( X_{i,s} = 1 \) if pattern \( s \) appears in \( t_i \) and \( X_{i,s} = 0 \) otherwise.

The association of pattern \( s \) and target class \( y = 1 \) can be investigated by conducting an independence test with the null hypothesis \( H_0: X_{i,s} \perp y \) via a \( 2 \times 2 \) contingency table as in Table 1. Here, \( n_D \) is the total number of transactions, and \( n_1 \) is the number of transactions with label \( y = 1 \). Moreover, \( n_s \) is the support for pattern \( s \); i.e., the number of transactions that contain pattern \( s \) and \( a_{s1} \) is the support for pattern \( s \) among class \( y = 1 \). A widely used independence test in
SSPM is Fisher’s exact test, which is a conditioned test in which $n_D, n_1, n_s$ are assumed to be fixed [23]. Under the null hypothesis of no association between $X_s$ and $y$, the count cell $a_s$ follows a hypergeometric distribution $P(a_s|n_D, n_1, n_s)$. Thus, the probability of observing the current table can be calculated as follows:

$$P(a_s|n_D, n_1, n_s) = \frac{\binom{n_1}{a_s} \binom{n_D-n_1}{n_D-a_s}}{\binom{n}{a_s}}.$$  

Consequently, the p-value, i.e., the probability of observing a contingency table that is equally or more extreme as the observed table under the null hypotheses, can be obtained as

$$p_{val} = \sum_{k=a_s}^{\min(n_D, n_s)} P(k|n_D, n_1, n_s).$$

If the p-value $p_{val} \leq \alpha$ holds for some significance level $\alpha$, we can reject the null hypothesis and conclude that pattern $s$ is associated with outcome $y$ under the significance level $\alpha$. The significance testing procedure controls the probability of falsely rejecting a true null hypothesis, i.e., the probability of falsely discovering a false pattern, under the desired significance level $\alpha$.

### 3.2 Multiple Testing

In the previous subsection, we described an individual test for a pattern. However, because we must consider many patterns, many hypotheses must be tested simultaneously. Suppose each test is conducted independently with a significance level of $\alpha$, the probability that at least one false discovery occurs would be much larger than $\alpha$. Typically, it is necessary to control the overall error of all tested hypotheses in such a case. Two widely used families are familywise error rate (FWER) and false discovery rate (FDR). In what follows, we focus on controlling the FWER, defined as the probability of making at least one false rejection. Assume that we want to test a hypotheses set $H = \{h_1, \ldots, h_{|H|}\}$. Letting $T \subseteq H$ be the subset of true null hypotheses and $R \subseteq H$ be the set of hypotheses that were rejected by the multiple-testing procedure, FWER is defined as follows:

**Definition 3.1 (Familywise error rate (FWER))**

$$\text{FWER} = \Pr[R \cap T \neq \emptyset]$$

The Bonferroni correction is the most straightforward method to control the FWER, which uses a corrected rejection threshold $\delta = \alpha/|H|$ for each test. However, when the number of hypotheses is large, the Bonferroni correction can become too conservative. Recently, in the context of SSPM, methods such as LAMP[7] have leveraged Tarone’s trick [2] to exclude untestable hypotheses that will never be significant. Specifically, if the test statistics are discrete, we can evaluate the minimal attainable p-value for that test. Let $\psi(h)$ be the minimal attainable p-value; hypothesis $h$ will never be rejected by a threshold $\sigma$ if $\psi(h) > \sigma$. In the case of Fisher’s exact test, the minimal attainable p-value $\psi(s)$ for a pattern $s$ is obtained as

$$\psi(s) = p(k|n_D, n_1, n_s) \; \text{where} \; k = \min\{n_1, n_s\}.$$  

Thus, if $\psi(s) > \delta$, the hypothesis related to pattern $s$ will never be significant for significance level $\delta$ regardless of the cell $a_s$. Moreover, in SSPM, $n_1$ can be considered fixed because the data are commonly given beforehand. Since $n_s = \sum X_{i,s}$, pattern $s$ can be ignored if there are too few transactions that contain pattern $s$, i.e., if the support for that pattern is too small. Because the number of hypotheses can be reduced using Tarone’s trick, a larger rejection threshold can be used, and more discoveries can be made. Formally, T-Bonferroni leverages Tarone’s trick by setting the rejection threshold $\sigma_{T-Bonferroni}$ as follows:

$$\sigma_{T-Bonferroni} = \max\{\sigma | \psi(\sigma) \leq \alpha\},$$

where $\psi(\sigma) = \{h | \psi(h) \leq \sigma\}$ is the testable hypothesis set regarding the rejection threshold $\sigma$.

### 3.3 Assumption and the Target Problem Class

Next, we introduce the proposed method’s key assumption and target problem class.

**Assumption 3.1 (P-Values Independence).** For any hypotheses pair $h_t \in T$ and $h_f \in F$, their p-values $p_{ht}$ and $p_{hf}$ are independent.

In other words, we assume that the p-values obtained from true and false patterns are mutually independent. Stated differently, we assume that the test statistics obtained from the true and false patterns are mutually independent, and the p-values are determined using these test statistics. We next claim that Assumption 3.3 holds for certain types of SSPM problems by the following proposition.

**Proposition 3.1.** In the setting of Sections 3.1 and 3.2, suppose p-values of Fisher’s exact test are considered. Given $D$, if for any two distinct patterns $s, s' \in S$, $\{t \in D | s \in t\} \cap \{t \in D | s' \in t\} = \emptyset$ holds, Assumption 3.1 holds.

**Proof.** Appendix A.7, available in the online supplemental material. \(\square\)

Proposition 3.3 claims that Assumption 3.3 can be satisfied with two conditions. First, Fisher’s exact test - a popular independent test in SSPM contexts, is used. Second, the pattern set is designed to separate the dataset into non-overlapping subsets which can be satisfied in several scenarios, for example

- Categorical dataset: consider a dataset with several categorical attributes $(x_1, x_2, \ldots, x_m)$ and a target label $y$. Assumption 3.3 holds by any pattern set $S \subseteq I_1 \times \cdots \times I_m$, where $I_d$ is the set of possible values for variable $x_d$.
- Transaction dataset with fixed transaction size: consider a dataset that for any transaction $t_i \in D$, the

| $y_i = 1$ | $X_{i,s} = 1$ | $X_{i,s} = 0$ | Total |
|---|---|---|---|
| $y_i = 0$ | $n_1 - a_s$ | $n_D - n_s - (n_1 - a_s)$ | $n_D - n_1$ |

**TABLE 1**

$2 \times 2$ Contingency Table for Pattern S
number of items in $t_i$ is $k$ for a fixed $k_i$, i.e., $\sum_{j \in I} x_{ij} = k$. Assumption 3.3 holds by any pattern set $S \subseteq I^k$.

As a counter-example, Assumption 3.3 does not hold for a set $S$ which contains both pattern $\{a\}$ and pattern $\{a, b\}$, since a transaction that contains $\{a, b\}$ will also contain $\{a\}$. It is noted that the categorical dataset setup fits most of our intended applications and is used for our real-world experiment in Section 7.

4 PROBLEM SETTING

To formally define our problem setting, we first introduce the concept of ordinal utility and clarify our goal of finding significant patterns with the highest utility. Next, we define a criterion for evaluating the goodness of the discovered result and discuss existing approaches’ limitations.

4.1 Ordinal Utility and the Dominating Subset

We focus on the ordinal utility between patterns, i.e., determining which of the two patterns is better. This kind of utility differs from the cardinal utility, which considers how good each pattern is. In SSPM, assigning a utility score for each pattern is not always practical for patterns with multiple items. By contrast, using the (partial) ordinal utility, we can easily define a flexible utility relationship between patterns, possibly using their items. We use $s_1 \succ s_2$ and $s_1 \simeq s_2$ to denote that “pattern $s_1$ is useful to pattern $s_2$” and “pattern $s_1$ is equally useful as pattern $s_2$”, respectively. Moreover, since each pattern $s$ is evaluated via a hypothesis test $h_s$ in SSPM, we similarly use $h_{s_1} \succ h_{s_2}$ and $h_{s_1} \simeq h_{s_2}$ to denote the ordinal utility between hypotheses. A preference order of utility for a pattern set $S$ is defined in the following.

Definition 4.1 (Preference order of utility). A preference order $\prec$ on a pattern set $S = \{s_1, \ldots, s_i, \ldots, s_k\}$ is a transitive binary relation in which $(S, \prec)$ is a partially ordered set.

Such a preference order can be predefined by the user. For example, suppose we are considering a dataset on medication usage in which each transaction includes a combination of drugs used by a patient and a binary class cured or not-cured. Moreover, we assume that we can define the total cost $\text{cost}(s)$ and the adverse effect level $\text{adv}(s)$ for each drug combination pattern $s$ as ordinal levels. A user who prefers patterns with a lower cost and less adverse effects can define the ordinal utility between patterns as

$$\text{cost}(s_1) = \text{cost}(s_2) \wedge \text{adv}(s_1) = \text{adv}(s_2) \iff h_{s_1} \simeq h_{s_2}$$

$$\text{cost}(s_1) < \text{cost}(s_2) \wedge \text{adv}(s_1) \leq \text{adv}(s_2) \Rightarrow h_{s_1} \succ h_{s_2}$$

$$\text{cost}(s_1) \leq \text{cost}(s_2) \wedge \text{adv}(s_1) < \text{adv}(s_2) \Rightarrow h_{s_1} \succ h_{s_2}.$$

We show an example of a pattern set $R = \{a, b, c, d, e, f\}$ in Fig. 2 where each point $a, \ldots, f$ represents the utility of that pattern, i.e., the cost and the adverse effect levels. The colored rectangle next to each point represents the area (patterns) that is less useful than that point (pattern). From the previous definition, we have that $h_b \prec h_d$ or $h_e \prec h_c$. We notice that pairs of patterns such as $h_b$ and $h_q$ in Fig. 2 are not comparable. In other words, we do not require the preference order $\prec$ to be complete. This design choice is especially useful because it can consider multiple aspects of the pattern that contains multiple items.

In SSPM, the utility of a pattern can be defined using the items included in that pattern. For example, in the medication example, let $\text{cost}(t)$ and $\text{adv}(t)$ be the ordinal level of the cost and the adverse effect level for a drug $t$. We can define a utility function $U : S \rightarrow N^d$ as

$$U(s) = \left( \frac{\text{cost}(s)}{\text{adv}(s)} \right) = \left( \frac{\sum_{i \in S} \text{cost}(t)}{\max_{i \in S} \text{adv}(s)} \right). \quad (6)$$

Next, we recall that our goal is to discover significant patterns with the highest utility. Hence, the utility of a discovered pattern set is determined only by the most high-utility patterns in that set. We call such a subset the (utility) dominating subset.

Definition 4.2 ((Utility) dominating subset). For a pattern set $K = \{s_1, \ldots, s_k\}$, we call $\text{dom}(K) \subseteq K$ the (utility) dominating subset of $K$ if $\text{dom}(K) = \{s \in K | \exists t \in K, s \prec s \}$

We also illustrate this concept by showing the dominating set $\text{dom}(R)$ for $R = \{a, b, c, d, e, f\}$ in Fig. 2 (right). We see that since $b \prec a$, we have $b \not\in \text{dom}(R)$. Similarly, we also have that $d \not\in \text{dom}(R)$ because $e \prec a$ and $f \not\in \text{dom}(R)$ because $f \prec d$. Finally, we obtain the dominating subset $\text{dom}(R) = \{a, c, d\}$ as these patterns are not comparable.

Our setting also requires that the discovered patterns must be true patterns. That is, letting the set of true patterns (the set of false null hypotheses) be $F$; our actual goal is to maximize the utility of true discoveries (rejected false hypotheses), i.e., the utility of $\text{dom}(R \cap F)$. Since $\forall R, \text{dom}(R \cap F) \subset \text{dom}(F)$, the best discovery result that can be achieved is $\text{dom}(F)$. However, in the SSPM setting, the number of discoveries is often limited because we have to control the number of false discoveries. Hence, it is often impossible to achieve $\text{dom}(F)$, especially when the number of true patterns is large. Thus, the practical goal in our setting is to discover a pattern set whose utility is as close to $\text{dom}(F)$ as possible. Moreover, because $F$ is unknown in a real-world setting, $\text{dom}(R)$ is adopted to evaluate the discovered patterns’ utility.

4.2 Criterion for Utility Evaluation

Here, we introduce a metric to compare the goodness of two pattern sets, or in other words, two rejected hypothesis sets. We will use this criterion to compare the utilities between discovered set $\text{dom}(R)$ and the optimal set $\text{dom}(F)$. We also use this criterion to compare the results of different methods for the real-world setting where $F$ is unknown.

Definition 4.3 (Utility measure). Given two pattern sets $K$ and $K'$, the utility measure from $K'$ to $K$ is denoted by $D_u(K||K')$.
4.3 Existing Approaches and Their Limitations

The most simple approach is to use standard multiple-testing procedures such as Bonferroni or Holm [24] to obtain significant patterns, then filter out less useful patterns. The limitation of this approach is obvious: they treat all hypotheses equally regardless of their utility. These methods are only developed to maximize the number of rejections \(|R|\) under the constraint of FWER. By contrast, our setting only focuses on the final dominating subset dom(\(R\)). Consequently, from the utility perspective, many rejections of less useful patterns in \(R\) are meaningless, resulting in a waste of the significance budget for controlling the type-I error of hypotheses that are not useful. Next, we discuss some alternatives that can leverage the utility of patterns.

4.3.1 Weighted Approaches

A straightforward way to utilize the pattern utility is using the weighted approach, such as the weighted Bonferroni procedure [25]. This procedure assigns a weight to each hypothesis based on its importance. More important hypotheses will have a higher chance of being rejected. Such importance can be obtained by quantizing the ordinal utility of each hypothesis, for example, by assigning the number of less preferred hypotheses (\(h\) = \(|\{h' \in H, h' > h\}|\)). However, if small weights are unfortunately assigned to false null hypotheses, many false null hypotheses will not be rejected because the rejection thresholds become too strict. In the worst case, the weighted approach would miss all the correct patterns. The weighted approach does not work well in practice because the weights must be assigned before the tests to control FWER properly, and no one knows which hypotheses are true null hypotheses.

4.3.2 Subfamily Multiple Testing for Hypotheses Stream

The hypotheses stream problem considers a stream of hypotheses sets \(H_1, H_2, \ldots\) which arrived in turn. The task is to reject one hypothesis for each hypotheses set while controlling the FWER. At each iteration \(t\), the procedure will reject or accept the hypothesis \(h_t^i\) with the smallest p-value in \(H_t\). To control the FWER such setting, Webb et al. proposed the Subfamily Multiple Testing procedure (SMT). SMT consumes a significant budget \(\Delta_t = p_t \times |H_t|\) for rejecting \(h_t^i\) in \(H_t\). They showed that the FWER in the hypotheses stream problem is controlled if the accumulated consumed budget is kept less or equal to the significant level \(\alpha\), i.e., \(\sum_t \Delta_t < \alpha\).

In adapting SMT to our setting, we introduce SMT-utility, which after rejecting a hypothesis \(h_t^i\), removes all the hypotheses that are less useful than the rejected hypothesis from the candidate set. Our setting can be considered a special case of SMT where the hypotheses set keeps shrinking, i.e., \(H_{t+1} \subset H_t\). We explored that in such a case, one can use a much less significant budget and still control the FWER, which results in more discoveries and higher utility.

4.3.3 Interactive Unmasking (i-FWER)

The recently proposed interactive testing procedure i-FWER by Dual Boyan et al. can also be adapted for our setting [21]. i-FWER is a multi-step testing procedure in which the user can use any prior information and the current result of the algorithm to remove unfavorable hypotheses while still controlling FWER. In the procedure of i-FWER, the p-value \(p_h\) of each hypothesis \(h\) is decomposed into two parts: masked p-value \(g(p_h)\) and missing bits \(f(p_h)\). Here, \(g: [0, 0.5] \rightarrow (0, 0.5)\) calculate a masked p-values \(\{g(p_h)\}_h\), e.g., \(g(p) = |p - 1\{p > 0.5\}|\), that provides some (masked) information about the true p-value \(p_h\). In each step, the user can use these masked p-values and any available side information to remove unfavorable hypotheses. On the other hand, \(f: [0, 1] \rightarrow \{-1, 1\}\) provides the missing bit of the masked p-values, e.g., \(f(p) = 2 \cdot 1\{p \leq 0.5\} - 1\), that are used to estimate the FWER of remaining hypotheses at each step. i-FWER keeps removing unfavorable hypotheses until this estimated FWER is first controlled by \(\alpha\). Finally, the procedure rejects only the remaining hypotheses whose missing bit \(f(p_h) = 1\).
In our setting, the side information is the ordinal utility between hypotheses and can be used to remove less preferred hypotheses in each step. Here, we introduce two ways to adapt i-FWER to our setting:

- **i-FWER-weight.** We define a score function \( f_{\text{score}}(h) = f_{\text{score}}(w(x_h), g(p_h)) \) to prioritize the utility while at the same time focusing on rejecting “promising” hypotheses. In our experiment, we employed the scoring function \( f_{\text{score}}(h) = (-g(p_h)) \cdot w(x_h) \). In each step, we remove the hypothesis with the smallest score.

- **i-FWER-utility.** We directly use the ordinal utility and the order of masked p-values \( \{g(p_h)\} \) to choose the hypothesis to be removed. In our setting, we note that the most unfavorable hypotheses are those that are both not significant and not useful. Thus, removing those less useful than “promising” hypotheses, i.e., the hypotheses likely to be rejected, is a valid strategy. Specifically, we focus on the hypotheses with the smallest masked p-values and choose to remove the most useless hypothesis that is less useful than these “promising” ones. Consider an iteration \( t \), without the loss of generality, let \( R_t = \{h_1, \ldots, h_{\lceil|H|-1\rceil}\} \) be the remaining hypotheses set at iteration \( t \) and assume that \( g(h_1) \leq g(h_2) \leq \cdots \leq g(h_{\lceil|H|-1\rceil}). \) Here, \( h_{\lceil|H|-1\rceil} \), whose masked p-value is the smallest, has the highest chance of rejection. If one or more hypotheses that are less useful than \( h_{\lceil|H|-1\rceil} \) exist, we remove the least useful one. If there is no such hypothesis, we shift our focus to \( h_{\lceil|H|-1\rceil} \) and continue the selection.

## 5 PROPOSED METHOD

### 5.1 The Proposed Method: SPUR

In our setting, after discovering a pattern set \( R_t \), discovering any pattern \( s \) that is less useful than a pattern in \( R_t \), i.e., \( \exists s' \in R, s' \succ s \), does not increase the result’s utility. Thus, once a hypothesis \( h_i \) is rejected, we can ignore all less useful hypotheses \( h_i \prec h_i \). Here, “ignoring a hypothesis” means “accepting the hypothesis”. Because we have to guarantee the type-I error when rejecting a hypothesis, we can save the significance budget by rejecting more useful hypotheses and, at the same time, accepting less useful ones.

We propose an iterative multiple-hypotheses procedure to conduct such an adaptive rejection process while controlling the FWER under Assumption 3.3. In particular, our method repeatedly (1) rejects the most significant hypothesis in the candidate hypothesis set and (2) ignores all hypotheses in the candidate set that are less useful than the rejected hypothesis in the last step. Next, we explain our algorithm along with the pseudo-code provided in Algorithm 1. We also note that FWER controlled by our adaptive procedure is not obvious, and we will discuss this in Section 6.1.

**Initialization and Notation.** First, we initialize the significance budget and the rejection set as \( \delta_1 = \alpha, R_0 = \emptyset \). Here, \( \delta_1 \) is used to control the FWER of the procedure at iteration \( t \). We define \( H_t \) as the set of the remaining candidate hypotheses at iteration \( t \) and \( H_t = H_t \). Furthermore, we also define \( \kappa_t(\sigma) = \{h \in H_t | \psi(h) \leq \sigma\} \) as the set of testable hypotheses at the \( t \)th iteration for a rejection threshold \( \sigma \).

#### Rejection Procedure. At each iteration \( t \), we decide whether to reject a hypothesis in \( H_t \) as follows

1. We obtain the smallest p-value \( p_t' = \min_{h \in H_t} p_h \) and the corresponding hypothesis \( h_t' = \arg\min_{h \in H_t} p_h \). We also assume that \( h_t' \) can be decided by a predefined rule if there is more than one hypothesis realizing the smallest p-value.
2. We obtain \( \sigma_t = \max \{\sigma : (\sigma - p_t'-1)|\kappa_t(\sigma)| \leq \delta_t\} \) as the rejection threshold for rejecting \( h_t' \).
3. If \( p_t' \leq \sigma_t \), we reject \( h_t' \) by adding it to the rejection set, i.e., \( R_{t+1} = R_t \cup \{h_t'\} \). We also ignore (accept) all less useful hypotheses in \( H_t \) by setting the next candidate set as \( H_{t+1} = \{h \in H_t | h \prec h_t'\} \). Finally, we modify the significance budget as \( \delta_{t+1} = \delta_t - \tau_t(p_t'-1) + p_t' \), where \( \tau_t = \delta_t/(\sigma_t - p_t'-1) \).
4. If \( p_t' > \sigma_t \), we stop and return the set \( R = R_{t-1} \).

#### Algorithm 1. SPUR

**Data:** Hypothesis Set \( H \), Statistical Level \( \alpha \)

**Result:** Reject set \( R \)

1. \( t := 1 \); \( R_0 := \emptyset \); \( H_1 := H_1 \); \( \delta_1 := \alpha \); \( p_0' := 0 \); \( R := \emptyset \);
2. while \( H_t \neq \emptyset \) do
3. \( \sigma_t := \max \{\sigma : (\sigma - p_{t'-1})|\kappa_t(\sigma)| \leq \delta_t\} \);
4. \( p_t' := \min_{h \in H_t} p_h \); \( h_t' := \arg\min_{h \in H_t} p_h \);
5. if \( p_t' \leq \sigma_t \) then
6. \( R_{t+1} := R_t \cup \{h_t'\} \); // reject
7. \( H_{t+1} := \{h \in H_t | h \prec h_t'\} \); // ignore
8. \( \tau_t := \delta_t/(\sigma_t - p_t'-1) \);
9. \( \delta_{t+1} := \delta_t - \tau_t(p_t'-1) + p_t' \); // modify
10. end
11. end
12. else
13. return \( R_{t-1} \);
14. end
15. \( t := t + 1 \);
16. end
17. return \( R_{t-1} \);

### 5.2 A Simple Case for Intuitive Understanding

In SPUR, the rejection threshold \( \sigma_t \) and significance budget \( \delta_t \) are managed in a way that is not so intuitive. We will prove that such updates are necessary to control the FWER properly in Section 6. Before providing the formal proof, we give a more intuitive account of our rejection rule by considering a simple case: the case without the Tarone’s trick, i.e., the case in which \( \forall h \in H : \psi(h) = 0 \). In such a case, the rejection rule of SPUR can be rewritten as

\[
p_t' \leq \alpha - \sum_{i=1}^{t-1} p_t'(|H_i| - |H_{i+1}| - 1) \Rightarrow \text{reject } h_t'
\]

otherwise \( \Rightarrow \) stop.

The detailed algorithm using this rejection rule is shown in Appendix B, available in the online supplemental material, and we have the following proposition.

**Proposition 5.1 (SPUR in the limited case).** If for any \( h \in H \), \( \psi(h) = 0 \), then the rejection sets obtained by Algorithm 1 and Algorithm 1 are identical.
Proof. Appendix A.8, available in the online supplemental material.

In fact, Algorithm 1 adopts a similar framework with SMT. Here, \( \delta_t \) can be interpreted as the remaining significant budget at step \( t \). The different Algorithm 1 and SMT is the significant budget \( \mu_t \) that costed for each rejection. In SMT, each rejection will cost a significant budget \( \Delta_{\text{SMT}}^t = \mu_t^t \). On the other hand, in the proposed method, rejecting \( h_t^r \) and ignoring less useful hypotheses than \( h_t^r \) will cost a significant budget \( \Delta_t = \mu_t^t[|H|] - |H_{t+1}|-1 \). We note that \( |H_t| - |H_{t+1}| - 1 \) is exactly the number of hypotheses that will be ignored due to the rejection of \( h_t^r \), i.e., \( |H| - |H_{t+1}| \) (line 8 of Algorithm 1). We also remark that when no hypotheses can be ignored, i.e., \( \forall i. \mu_t^t[|H|] - |H_{t+1}| - 1 = 0 \), our method reduces to the step-down Holm method[24]. The full SPUR procedure is obtained by leveraging the Tarone’s trick to additionally consider the testability of the candidate hypotheses set at each iteration.

6 THEORETICAL ANALYSIS

6.1 FWER Guarantee

Here, we show that the proposed method can control the FWER properly under Assumption 3.3 by Theorem 6.1.

Theorem 6.1 (FWER controlling). Under Assumption 3.3, the proposed SPUR procedure conducted with significance level \( \alpha \) controls the familywise error rate at \( \alpha \).

We first define the stochastic process obtained by the proposed method SPUR to show this theorem.

Definition 6.1 (SPUR process). A SPUR (stochastic) process is a stochastic process \( \{X_t\}_{t=1}^{T_{\text{SPUR}}} \) stopped at stopping time \( T_{\text{SPUR}} \) where \( X_t = (\delta_t, R_t, p_t^r, h_t^r, \sigma_t, \tau_t) \) are obtained from the algorithm SPUR and \( T_{\text{SPUR}} = \min\{t \in [|H|] : p_t^r > \alpha_t \} \).

We remark that since we can obtain \( H_t = \{h \in [H] \mid h \not\in R_{t-1}, h \not\in h_t^r \} \), we do not need to include \( H_t \) in \( X_t \).

Using the stochastic process \( \{X_t\}_{t=1}^{T_{\text{SPUR}}} \), we next rewrite the event of rejecting at least one true hypothesis, i.e., the event of occurring a Type-I error.

Lemma 6.1. Consider a SPUR process \( \{X_t\}_{t=1}^{T_{\text{SPUR}}} \), let \( E_t = \{h_t^r \in T, p_t^r \leq \alpha_t \} \) and \( T_E = \min\{t \in [|H|] : t \leq T_{\text{SPUR}} \land E_t \} \), the following holds

\[ R \cap T \neq \emptyset \Rightarrow \{T_E \leq |H|\} \]

Proof. Appendix A.2, available in the online supplemental material.

Here, \( E_t = \{h_t^r \in T, p_t^r \leq \alpha_t \} \) is the event of rejecting a true hypothesis at the \( t \)-th iteration and \( T_E = \min\{t \in [|H|] : t \leq T_{\text{SPUR}} \land E_t \} \) is the first iteration that a true hypothesis is rejected. Lemma 6.1 claims that the FWER is actually the probability of the first Type-I error occurring at some step \( t_0 \leq |H| \). To show the FWER controlling, we have to consider the relationship between the p-values of the true and false hypotheses and the rejection threshold at each iteration. However, such dependencies are complicated in the SPUR process \( \{X_t\}_{t=1}^{T_{\text{SPUR}}} \). Thus, we instead consider an alternative stochastic process that only depends on the false hypotheses and show the FWER controlling by analyzing this process. In addition, we illustrate the relation of entities in the SPUR process using a graph shown in Fig. 4.

Definition 6.2 (False hypotheses based process). A false hypotheses based process is a stochastic process \( \{Y_t\}_{t=1}^{T_{\text{FALSE}}} \) stopped at stopping time \( T_{\text{FALSE}} \) where \( Y_t = (\delta_t, R_t, p_t^r, h_t^r, \sigma_t, \tau_t) \) are obtained from the algorithm SPUR with the following modification

\[ p_t^r = \min_{h \in H_t \cap F} p_h \]

Next, we define a sequence of true hypotheses’ p-values \( \{p_t^t\}_{t=1}^{T_{\text{FALSE}}} \) where each \( p_t^t \) is obtained using the value \( Y_t \) of the false hypotheses based process.

Definition 6.3 (Alternative true hypotheses sequence). A set of r.v. \( \{p_t^t\}_{t=1}^{T_{\text{FALSE}}} \) is said to be an alternative true hypotheses sequence obtained from the false hypotheses based process \( \{Y_t\}_{t=1}^{T_{\text{FALSE}}} \) if for \( t \leq T_{\text{FALSE}} \)

\[ p_t^t = f(Y_t, \{p_h\}_{h \in T}) = \min_{h \in H_t \cap F} p_h. \]

Since SPUR rejects the most significant hypothesis \( h_t^r \) at each step, the event \( \{h_t^r \in T \} \) (and \( \{h_t^r \in F \} \)) depends on the comparison of \( \min_{h \in H_t \cap F} p_h \) and \( \min_{h \in H_t \cap R} p_h \). We next consider this comparison using the false hypotheses based process and the alternative true hypotheses sequence, while claims its relation to Lemma 6.1.

Lemma 6.2. Consider \( \{Y_t\}_{t=1}^{T_{\text{FALSE}}} \) and \( \{p_t^t\}_{t=1}^{T_{\text{FALSE}}} \) as defined in Definition 6.1 and 6.1. Let \( E_t = \{p_t^t \leq p_t^r, p_t^t \leq \sigma_t^t, T_E = \min\{t \in [|H|] : t \leq T_{\text{FALSE}} \land E_t \} \) and \( T_E \) as defined in Lemma 6.1

\[ T_E \geq T_t^* \text{ almost surely.} \]

Proof. Appendix A.4, available in the online supplemental material.

We give an illustration on the event \( E_t \) and its relationship with other entities in Fig. 5. From Lemma 6.1, we have that

\[ \Pr[R \cap T \neq \emptyset] \leq \Pr[T_E \leq |H|] \leq \Pr[T_{\text{F}} \leq |H|]. \]

Moreover, because \( \Pr[T_{\text{F}} \leq |H|] = E_{\{p_h\}_{h \in F}} \Pr[T_{\text{F}} \leq |H|]\{p_h\}_{h \in F}] \), we will next find the upper bound of \( \Pr[T_{\text{F}} \leq |H|]\{p_h\}_{h \in F}] \).

Lemma 6.3. Using the same definition of Lemma 6.1 and let \( k = T_{\text{F}} \) under Assumption 3.3, the following holds

\[ T_{\text{F}} \geq T_t^* \text{ almost surely.} \]
Appendix A.6, available in the online supplemental material. Let \( T \) be the false hypotheses set, \( R^{\tau}_{\text{False}} \) be the rejection set by T-Bonferroni; then \( R \cap F \geq R^{\tau}_{\text{False}} \cap F \).

Proof. Appendix A.10, available in the online supplemental material.

Theorem 6.3 (Comparison with SMT in the limited case). For arbitrary preference order \( \prec \), assuming that \( h \in H \), \( \psi(h) = 0 \), let \( F \) be the false hypotheses set, \( R \) be rejected by SPUR and \( R^\tau_{\text{SMT}} \) be the rejection set by SMT; then \( R \cap F \geq R^\tau_{\text{SMT}} \cap F \).

Proof. Appendix A.11, available in the online supplemental material.

These guarantees of utility improvement are the most critical advantage of SPUR compared to other approaches. As discussed in Section 4.3, the discovery result by weighted approaches can be heavily affected if the false hypotheses are assigned small weights. In the worst case, the weighted approach would be inferior to methods that do not consider utility, e.g., T-Bonferroni. In contrast, although the merit of SPUR also decreases when the utility of the false hypotheses is low, it always guarantees a utility that is not less useful than T-Bonferroni.

7 Experiment

This section evaluates SPUR using a synthetic experiment and three real-world datasets.

7.1 Synthetic Experiment I

We conduct the first synthetic experiment with two goals. First, we verify if SPUR can adequately control FWER. Particularly, we demonstrate that adjusting the significance budget by \( \delta_{i+1} = \delta_i - \tau_i (p_i - p_{i-1}) + p_i \) is necessary to control the FWER by showing that letting \( \delta_{i+1} = \delta_i \) would violate the FWER. Second, we verify if SPUR can correctly reject hypotheses with higher utility compared to other methods. We confirm our discussion in Section 6.2 about the limitation of the weighted approaches.

7.1.1 Experiment Setting

We consider a set of 150 hypotheses \( H = \{h_1, h_2, \ldots, h_{150}\} \), where 10 of them are false null hypotheses. Let \( F \) be the set of false hypotheses and let the preference be \( h_i > h_j \) for \( i < j \), i.e., the hypothesis with a smaller index be more useful. We illustrate our setting in Fig. 6. For each hypothesis \( h_i \), we generate a dataset \( D_i = \{x_{ij}\}_{j=1}^{20} \) with 20 samples. Since we do not consider the Tarone’s trick in this experiment, we adopt the z-test and set up the null hypothesis \( H_0 : \mu = 0 \) for all hypotheses. For the true null hypotheses, i.e., for \( h_i \in T \), we generate data using the distribution \( \mathcal{N}(\mu_0, 0) \).

Although using different mechanisms, methods that consider the utility typically can reject more false hypotheses if the utility of these false hypotheses is high and results in a higher utility \( u(R) \). In contrast, these methods might struggle when the utility of the false hypotheses is low as fewer false hypotheses can be rejected and the utility \( u(R) \) decreases. To investigate such change in performance, we consider five settings in which we gradually decrease the usefulness of the false hypotheses set, i.e., by changing the indexes of false hypotheses in five settings.
We named five settings by how useful the false hypotheses set is compared to true hypotheses.

1) Very high: \( F = \{h_5, h_{10}, \ldots, h_{45}, h_{50}\} \)

2) High: \( F = \{h_{25}, h_{30}, \ldots, h_{50}, h_{75}\} \)

3) Medium: \( F = \{h_{50}, h_{55}, \ldots, h_{95}, h_{100}\} \)

4) Low: \( F = \{h_{75}, h_{80}, \ldots, h_{120}, h_{125}\} \)

5) Very low: \( F = \{h_{100}, h_{105}, \ldots, h_{145}, h_{150}\} \)

Moreover, we also consider different values of \( \mu_i \), i.e., the mean of the false null distribution. By alternating \( \mu_i \), we aim to alternate how much the utility is considered in selecting rejection hypotheses. Specifically, these methods (excluding Bonferroni) consider each hypothesis by two aspects: the p-value, i.e., how significant that hypothesis is, and the utility, i.e., how preferable that hypothesis is. When the difference between \( \mu_1 \) and \( \mu_0 \) is small, the difference between the p-values of true and false hypotheses is likely to be small. Thus, the utility will play a large part in selecting hypotheses. Meanwhile, when the difference between \( \mu_1 \) and \( \mu_0 \) is large, the p-value will play a more important role in selecting hypotheses. We consider the following three values of \( \mu_1 \in \{0.5, 0.75, 1.0\} \).

For each setting, we generated 10,000 datasets, applied each generated dataset to the six methods, and then evaluated the FWER and the utility of rejected set. We remark that the p-values obtained in this setting satisfy Assumption 3.3 although this setting does not employ Fisher’s exact test.

### 7.1.2 Comparative Methods and Criterion

#### Comparison Methods.

Including the proposed method SPUR, we use the following six methods for comparison.

- **SPUR**: the proposed method SPUR.
- **Bonferroni**: T-Bonferroni method with \( \sigma = \alpha/|H| \).
- **w-Bonferroni**: the weighted Bonferroni with weighting function \( w(h_i) = |H, h < h_i| + 1 \).
- **SMT-utility**: SMT method in which the next iteration’s hypotheses set is obtained similarly to SPUR (Section 4.3.2).
- **i-FWER-weight** and **i-FWER-utility**: two adaptations of i-FWER to the proposed setting (Section 4.3.3).
- **invalid-SPUR**: the version of SPUR where \( \delta_{i+1} = \delta_i \).

#### Evaluation Criterion.

In this synthetic experiment, \( Dom(R \cap F) \) can be computed because \( F \) is known and we adopt it to evaluate the rejected hypotheses’ utility. Since our utility is ordinal, we first define \( r(h) \) as the utility ranking of a false hypothesis \( h \) among all 10 false hypotheses. For example, in the medium utility setting, we have \( r(h_{50}) = 10, r(h_{55}) = 9, \ldots, r(h_{100}) = 1 \). The larger \( r(h) \), the more useful that false hypothesis is. We emphasize that this ranking is obtained by comparing the utility between only false hypotheses. Then, the utility \( u(R) \) of a rejected set \( R \) is obtained by the utility ranking of the most useful false hypothesis that got rejected. i.e., \( u(R) = \max_{h \in R \cap F} r(h) \). When no false hypothesis is rejected, i.e., when \( R \cap F = \emptyset \), we assign \( u(R) = 0 \).

#### Evaluating the Utility of the Rejected Set.

For each rejection, we evaluate the following metrics:

- **Evaluating the FWER Controlling.** The FWER and the average number of rejections according to five methods are given in Table 2. From the table, we observed that all the methods successfully controlled the FWER in all five settings. In contrast, invalid-SPUR failed to control the FWER for the medium utility setting, even though its number of rejections is relatively small. This result demonstrates that the management of the significance budget \( \delta_i \) of SPUR (line 10 of Algorithm 1) is necessary for controlling the FWER. Although the violation in this experiment is not that significant, one can find more results where invalid-SPUR failed hardly in our early version of this work[3].

#### The Limitation of Weighted Approaches.

We first observed that the average ranking by Bonferroni does not change when the usefulness of false hypotheses changes. Because Bonferroni does not consider the utility between hypotheses, each false hypothesis always has an equal chance of being rejected, regardless of its utility. For the methods that consider the utility, we observed that the utility of the rejected set decreases when the usefulness of the false hypotheses decreases.

| Setting       | Very high | High   | Medium | Low    |
|---------------|-----------|--------|--------|--------|
| SPUR          | 0.008     | 0.046  | 0.051  | 0.049  | 0.048  |
| Bonferroni    | 0.044     | 0.043  | 0.047  | 0.045  | 0.045  |
| SMT-utility   | 0.007     | 0.041  | 0.042  | 0.041  | 0.037  |
| i-FWER-weight | 0.050     | 0.046  | 0.051  | 0.050  | 0.050  |
| i-FWER-utility| 0.046     | 0.046  | 0.051  | 0.048  | 0.048  |
| w-Bonferroni  | 0.041     | 0.043  | 0.049  | 0.047  | 0.046  |
| Invalid SPUR  | 0.008     | 0.050  | 0.053  | 0.050  | 0.049  |

#### 7.1.3 Results and Discussion

- **Evaluating the FWER Controlling.** The FWER and the average number of rejections according to five methods are given in Table 2. From the table, we observed that all the methods successfully controlled the FWER in all five settings. In contrast, invalid-SPUR failed to control the FWER for the medium utility setting, even though its number of rejections is relatively small. This result demonstrates that the management of the significance budget \( \delta_i \) of SPUR (line 10 of Algorithm 1) is necessary for controlling the FWER. Although the violation in this experiment is not that significant, one can find more results where invalid-SPUR failed hardly in our early version of this work[3].

- **Evaluating the Utility of the Rejected Set.** On the left side of Fig. 7, we show each methods’ average utility \( u(R) \) across 10,000 runs. We show the number of times each method achieved the highest utility among the five methods on the right side. To be specific, in each run (using the same data), we compare the utility \( u(R) \) of the rejection set by each method and decide which method (methods) rejected the set with the highest utility. In this manner, we count in 10,000 runs how many times a method is the best among the five methods.

We first observed that the average ranking by Bonferroni does not change when the usefulness of false hypotheses changes. Because Bonferroni does not consider the utility between hypotheses, each false hypothesis always has an equal chance of being rejected, regardless of its utility. For the methods that consider the utility, we observed that the utility of the rejected set decreases when the usefulness of the false hypotheses decreases.

- **The Limitation of Weighted Approaches.** First, we observed that the utility of weighted methods w-Bonferroni and i-FWER-weight drops when the usefulness of false hypotheses decreases. This decrease in utility can be explained by the dependence between the assigned weights and the rejection threshold. When the false hypotheses have high utility, weighted methods work well because large weights are

| Setting       | Very high | High   | Medium | Low    |
|---------------|-----------|--------|--------|--------|
| SPUR          | 0.008     | 0.046  | 0.051  | 0.049  | 0.048  |
| Bonferroni    | 0.044     | 0.043  | 0.047  | 0.045  | 0.045  |
| SMT-utility   | 0.007     | 0.041  | 0.042  | 0.041  | 0.037  |
| i-FWER-weight | 0.050     | 0.046  | 0.051  | 0.050  | 0.050  |
| i-FWER-utility| 0.046     | 0.046  | 0.051  | 0.048  | 0.048  |
| w-Bonferroni  | 0.041     | 0.043  | 0.049  | 0.047  | 0.046  |
| Invalid SPUR  | 0.008     | 0.050  | 0.053  | 0.050  | 0.049  |
to some extent, is large. Thus, using
\[ \mu_1 \]
and
\[ \mu_2 \]
are assigned for the false hypotheses. In contrast, small weights are assigned when the false hypotheses have low utility. For the latter, many false hypotheses could not be rejected because their rejection thresholds became too strict. This phenomenon can also be confirmed by looking at the decrease in the number of rejections in Table 3.

We observed that the gap between the weighted approaches and SPUR in the very high setting decreases when \( |\mu_0 - \mu_1| \) increases. This is because the rejection decisions of weighted methods for small \( |\mu_0 - \mu_1| \) rely more on the utility weights than for large \( |\mu_0 - \mu_1| \). Particularly, when \( |\mu_0 - \mu_1| \) is small (i.e., \( \mu_1 = 0.5 \) and \( \mu_1 = 0.75 \)), the p-values obtained from the true null hypotheses, i.e., \( N(\mu_0 = 0, \sigma = 1) \), and the false null hypotheses, i.e., \( N(\mu_1, \sigma = 1) \), become less distinguishable. Hence, the utility weights dominate the decision of which hypotheses to reject. In the "very high" setting, because large weights are set for false hypotheses, weighted methods can reject many false hypotheses, resulting in better discoveries than SPUR. However, when \( |\mu_0 - \mu_1| \) increases, i.e., \( \mu_1 = 1 \), the contribution of the p-values to the rejection decisions increases, and the gap between weighted methods and SPUR vanishes. Moreover, if the utility weights of the false hypotheses are not large enough, i.e., other than "very high", weighted approaches’ performance drops drastically, especially for small \( \mu_1 \).

The Problem of i-FWER. Although the i-FWER-utility method also achieved a high utility in the very high setting, its performance worsens when the false hypotheses’ usefulness decreases. The problem of i-FWER-utility is that, in the middle of the procedure where the selection for removal is made, only the masked p-value \( g(p_h) \) is available. Although \( g(p_h) \) provides the information of the real p-value \( p_h \) to some extent, it does not guarantee that a hypothesis with a small \( g(p_h) \) (even the smallest) will be rejected. The final decision for any rejection in i-FWER is only made at the end of the procedure when the remaining bits \( f(p_h) \) are revealed. In other words, there is a chance that \( g(p_h) \) is small, but \( p_h \) is large. Thus, using \( g(p_h) \), i-FWER-utility can accidentally remove useful false hypotheses. It is especially problematic when the utility of the false hypotheses is not high, as their chance of being removed by mistake increases. Moreover, the i-FWER-weight method, besides the limitation of the weighted approach, also inherits this problem of the i-FWER framework. That is why i-FWER-weight was the best in the very high setting but quickly became the worst in the other settings.

Utility Guarantee by SPUR. We can see that SPUR always achieved a rejection set that is not less useful than Bonferroni and SMT-utility in all the settings. This result follows our utility guarantee in Theorem 6.2 and Theorem 6.2. We note that other utility-oriented approaches only perform well when the usefulness of false hypotheses is very high. Otherwise, these methods can fail and perform worse than T-Bonferroni. Because the preference and the weights must be assigned beforehand, and no one knows which hypotheses are true, these approaches do not necessarily work well in practice. Meanwhile, although our method might not be the best in a few scenarios, it offers a stable solution that always performs equally or better than the gold-standard method T-Bonferroni.

We also observed that SPUR could achieve a rejection set with a higher utility if the false hypotheses have a higher utility. When the utility of the false hypotheses is high, SPUR can ignore many non-useful hypotheses once a false hypothesis gets rejected. Thus, SPUR can save its significant budget for later rejections, leading to a rejected set with higher utility.

### 7.2 Synthetic Experiment II

Next, we further investigate when and how SPUR outperforms SMT-utility. In Fig. 7, the behavior of SMT-utility is the most similar to our method compared to other methods. This is because SMT-utility uses the same strategy for

| Setting     | Very High | High     | Medium   | Medium   | Low      |
|-------------|-----------|----------|----------|----------|----------|
| SPUR        | 2.844     | 2.631    | 2.571    | 2.474    | 2.395    |
| Bonferroni  | 5.535     | 5.566    | 5.575    | 5.581    | 5.540    |
| SMT-utility | 2.827     | 2.597    | 2.526    | 2.427    | 2.342    |
| i-FWER-utility | 2.800 | 2.391    | 2.290    | 2.186    | 2.104    |
| i-FWER-weight | 6.229 | 4.548    | 4.005    | 3.635    | 3.374    |
| w-Bonferroni | 6.272 | 5.966    | 5.594    | 5.015    | 3.973    |
| Invalid SPUR | 2.852 | 2.641    | 2.581    | 2.482    | 2.402    |

### Table 3

Average Number of Rejections \(|R| (\mu_1 = 1)\)
removing hypotheses and is only different from SPUR in the management strategy of the significant budget.

### 7.2.1 Experiment Setting
We consider 150 hypotheses, in which 25 of them are false hypotheses and are distributed evenly, i.e., \( F = \{ h_5, h_{10}, \ldots, h_{145}, h_{150} \} \). Similar to the previous experiment, we adopt the z-test and set up the null hypothesis \( H_0 : \mu = 0 \) for all hypotheses. The data are sampled from \( \mathcal{N}(\mu_0 = 0, \sigma = 1) \) and \( \mathcal{N}(\mu_1 = 0.75, \sigma = 1) \) for true null and false hypotheses, respectively. We alternate the maximal number of hypotheses that can be removed in each iteration, i.e., \( (|H_t| - |H_{t+1}| - 1) \) in each step, and compare the utility of their rejected set. For example, let the maximal number of removals be ten hypotheses. Then, even if more than ten hypotheses can be removed in step \( t \), we only remove the least useful ten hypotheses. We also adopt the same evaluation criteria as the previous experiment.

### 7.2.2 Results and Discussion
In Fig. 8, we showed the results by SPUR and SMT-utility along with Bonferroni after 10,000 runs. We first observed that both methods performed well when many hypotheses are allowed to be removed in each iteration, and their performance dropped when fewer removal are allowed. Interestingly, from the relative comparison on the right side of Fig. 8, we observed that the (relative) difference in performance between SPUR and SMT-utility increases when the allowed number of removals decreases. We remark that for each rejection, SMT-utility consumes a budget \( \Delta_t^{SMT} = p_t[H_t] \) while SPUR consumes a budget of \( \Delta_t = p_t(|H_t| - |H_{t+1}| - 1) \). When the number of removals \( (|H_t| - |H_{t+1}| - 1) \) in each step gets smaller, \( \Delta_t \) also becomes smaller while \( \Delta_t^{SMT} \) stays unchanged. When the difference in budget consumption between the two methods increases, SPUR can discover more useful patterns and thus result in better utility than SMT. Also, from Fig. 8, we observed that when the number of removals decreases, SMT-utility eventually worsens than Bonferroni. On the other hand, SPUR did not worsen than Bonferroni, even when no removal is allowed. In SSPM problem with partially ordered preference, the number of possible removals is typically much smaller than the presented setting.

### 7.3 Real-World Datasets
We conduct real-world data experiments to confirm if SPUR can discover more useful patterns in a real-world situation where the Tarone’s trick must be considered. We adopted the three datasets Adult[28], Crash Report[29], and Crime[30] for five mining tasks.

#### 7.3.1 Data Pre-Processing and Task Setup
All three datasets consist of some explanatory variables and a target class. To elaborate, the Crash Report dataset[29] is a dataset of traffic accidents, including information such as speed limits, weather or light conditions, and a class \( y = 1 \) indicates if it is an injury or fatal crash. The Crime dataset[30] consists of criminal records with information about the place, street, and time along with a class for the crime category: against a person, property, or society. We set up three tasks for this dataset and used the category as the target class. Finally, the Adult dataset[28] contains several demographic attributes and a class \( y = 1 \) of income \( > 50K \) for different individuals.

To set up the candidate pattern set and the null hypotheses in each dataset, we focused on some variables and defined the mining task based on these variables. We translate the values of these variables into items by categorizing them with predefined rules. For example, for the hours-per-week attribute in the Adult dataset is categorized into \( < 20 \), \( [20 - 29] \), \( \ldots, [\geq 60] \). A valid pattern is a combination of all explanatory variables, with one item for each variable. This setup guarantees no samples overlap between patterns, and we adopted the Fisher’s exact test for all five tasks (one-sided for the Crash and Adult datasets, two-sided for the Crime dataset). Our goal is to discover the combinations of levels significantly associated with the target class.

Next, we define the ordinal utility between patterns. We first divide the explanatory variables into utility and family variables. Specifically, we state \( \mathcal{K} \supset \mathcal{K}' \) if their family variables are identical, and the utility variables of \( \mathcal{K} \) are more useful than \( \mathcal{K}' \). For example, for the Adult dataset, we let (sex, work-class, and occupation) be the family variables and hours-per-week, education be the utility variables. Here, based on prior knowledge that “a higher income correlates with higher education level and more work hours”, we oppositely prefer patterns with lower education level and fewer work hours. That is, we aim to discover patterns that are unexpectedly associated with the class \( > 50K \). The family variables are effective in finding useful patterns for various combinations of (sex, work-class, and occupation).

In addition, in the Crash task, we look for the patterns that sound safe but are unexpectedly associated with an injury accident. For example, for speed-limit, we prefer patterns with a lower speed limit. For weather, we most prefer clear, then cloudy and least prefer bad weather. In the Crime task, assuming that crime tends to happen at midnight rather than daylight, we prefer patterns in which the crime happens at midday. We fully describe the preprocessing procedure for the three real datasets in Appendix C, available in the online supplemental material.

#### 7.3.2 Results and Discussion
Table 4 shows the number of discoveries \( |R_S| \) by SPUR, \( |R_T| \) by T-Bonferroni, and the utility measure \( D_s(R_S||R_T) \), for
each significance level $\alpha$. We do not show $D_u(R_T|R_S)$, because $D_u(R_T||R_S) = 0$ for all settings, i.e., because SPUR always discovered a pattern set that is not less useful than T-Bonferroni. Especially, $D_u$ is large for the Crash, Property, and Society tasks, indicating that SPUR discovered many high-utility patterns that T-Bonferroni cannot discover.

Meanwhile, the utility measure $D_u(R_S||R_S)$ is comparatively small for the Adult and Person tasks. We explain this by considering T-Bonferroni’s number of discovered patterns $|R_T|$. First, we observed that $|R_T|$ in these tasks is smaller than those of other tasks. Moreover, even when increasing the significance level $\alpha$, only a few additional patterns are discovered. In such a case, since the number of patterns that can be discovered is small in the first place, even if SPUR can achieve a larger rejection threshold by ignoring less useful hypotheses, the number of newly discovered patterns would not increase. In other words, SPUR performed especially well for cases such as the Crash, Property, and Society. We explain this as T-Bonferroni did not discover many useful true patterns due to the FWER constraint, while there are many useless patterns in $R_T$.

Next, we focus on the discovery process of two methods using the Adult task. In Fig. 1, we show the discoveries by two methods with $\alpha = 0.05$ for two families (Male, Self-emp, Prof-specialty) and (Male, Private, Exec-managerial). We also show the sorted indexes of $p$-values for discovered patterns. As we can see from Fig. 1, by considering both the significance and the utility of discovered patterns, SPUR can efficiently expand its dominance of patterns with smaller number of rejections. By contrast, without considering utility, T-Bonferroni wastes its significant budget by rejecting less useful hypotheses. As a result, SPUR can discover more useful patterns that T-Bonferroni fails to discover. This advantage is not limited to discovering patterns in the same family. However, it is also helpful in discovering patterns of other families, as we observed several pattern families that only SPUR could discover.

In Fig. 1, SPUR discovered that a male (private) executive/manager who works 40–50 hours a week is likely to have an income > 50K even if he has just graduated from high school. Meanwhile, the required education level found by T-Bonferroni is to graduate from a college. Moreover, a male who graduated from a professional school and is working in a (self-employed) professional specialty is likely to have an income of > 50K even if he only works for 30–40 hours a week, while the requirement discovered by T-Bonferroni is 40–50 hours a week. Other than those patterns, in the Crash dataset, SPUR discovered many patterns that seemed safe but are still associated with an accident and injury, which T-Bonferroni cannot discover. In this way, SPUR can discover significant patterns and, simultaneously, be preferred by the user.

8 CONCLUSION

This study focused on the utility of patterns in the SSPM context. We introduced the problem of discovering statistically significant patterns with the highest utility, giving the ordinal utility between patterns. We proposed a novel iterative multiple-testing framework, SPUR, that alternately rejects a hypothesis and safely ignores hypotheses that are less useful than the rejected one. We theoretically showed that SPUR controls the FWER guarantee under a particular assumption and always guarantees an equal or better utility than T-Bonferroni and SMT. Finally, we conducted several experiments with synthetic and real-world datasets to demonstrate that the proposed method can discover more useful patterns under the constraint of type-I error.

Acknowledgments

We are also grateful to Professor Ichiro Takeuchi for providing useful discussions and advices.

References

[1] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, V. S. Tseng, and P. S. Yu, “A survey of utility-oriented pattern mining,” ACM Trans. Knowl. Data Eng., vol. 33, no. 4, pp. 1306–1327, Apr. 2021.
[2] R. E. Tarone, “A modified bonferroni method for discrete data,” Biometrics, vol. 46, no. 2, pp. 515–522, Jun. 1990.
[3] T. Q. Tran, F. Fukushima, Y. Akimoto, and J. Sakuma, “Statistically significant pattern mining with ordinal utility,” in Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2020, pp. 1645–1655.
[4] W. Hämäläinen, “Kingfisher: An efficient algorithm for searching for both positive and negative dependency rules with statistical significance measures,” Knowl. Inf. Syst., vol. 32, no. 2, pp. 383–414, Aug. 2012.
[5] G. I. Webb, “Discovering significant patterns,” Mach. Learn., vol. 68, no. 1, pp. 1–33, Jul. 2007.
[6] R. Agrawal, “Fast algorithms for mining association rules,” in Proc. 20th Int. Conf. Very Large Data Bases, 1994, pp. 487–499.
[7] A. Terada, K. Tsuda, and J. Sese, “Fast and memory-efficient significant pattern mining via permutation testing,” in Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2015, pp. 725–734.
[8] F. Linares-López, M. Sugiyama, L. Papaxanthos, and K. Borgwardt, “Fast and memory-efficient significant pattern mining via permutation testing,” in Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2015, pp. 725–734.
[9] S.-I. Minato, T. Uno, K. Tsuda, A. Terada, and J. Sese, “A fast method of statistical assessment for combinatorial hypotheses based on frequent itemset enumeration,” in Proc. Mach. Learn. Knowl. Discov. Databases, 2014, pp. 422–436.
[10] T. Pellegrina, M. Riondato, and F. Vandin, “SpuManTE: Significant pattern mining with unconditional testing,” in Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2019, pp. 1528–1538.
[11] J. Komiyama, M. Ishihata, H. Arimura, T. Nishiyabashi, and S.-i. Minato, “Statistical emerging pattern mining with multiple testing correction,” in Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Mining, 2017, pp. 897–906.
Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.