


SPuMANTE: Significant Pattern Mining with Unconditional Testing

Anonymous Author(s)

VSQPRD: *Vino Spumante di Qualità Prodotto in Regione Determinata*
(Italian: *Quality Sparkling Wine Produced in Determined Region*)

ABSTRACT

We present SPuMANTE, an efficient algorithm for mining significant patterns from a transactional dataset. SPuMANTE ensures that the probability of reporting one or more false discoveries is less than an user-specified threshold (i.e., SPuMANTE controls the Family-wise Error Rate). A key ingredient of SPuMANTE is UT, our novel unconditional statistical test for evaluating the significance of a pattern, that requires fewer assumptions on the data generation process and is more appropriate for a knowledge discovery setting than classical conditional tests, such as the widely used Fisher's exact test. Computational requirements have limited the use of unconditional tests in significant pattern discovery, but UT overcomes this issue by taking computing the required probabilities in a novel efficient way. SPuMANTE combines UT with recent results on the supremum of the deviations of pattern frequencies from their expectations, founded in statistical learning theory. This combination allows SPuMANTE to be very efficient, while also enjoying high statistical power. The results of our experimental evaluation show that SPuMANTE allows the discovery of statistically significant patterns while properly accounting for uncertainties in patterns' frequencies due to the data generation process.

 **NOTE FROM MATTEO (2/13):** The result sentence feels weak.

CCS CONCEPTS

• **Mathematics of computing** → Contingency table analysis; • **Information systems** → Data mining.

KEYWORDS

Family-wise Error Rate, Hypothesis Testing, Itemset Mining

ACM Reference Format:

Anonymous Author(s). 2019. SPuMANTE: Significant Pattern Mining with Unconditional Testing. In *KDD '19: The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 4–8, 2019, Anchorage, AK, USA*. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Significant Pattern Mining [10] is a variant of *Frequent Pattern Mining* [1], in which transactions have binary class labels and the

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
KDD'19, August 4–8, 2019, Anchorage, AK, USA
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

objective is to identify patterns with a *statistically significant association* to one of the two labels, and finds applications in a wide range of domains, including market basket analysis, social networks, and molecular medicine. In these domains, finding *reliable* associations is crucial. Significant patterns supply different information than frequent ones (i.e., appearing in a fraction at least θ of all the transactions of a dataset), providing important insights in many applications: for example, in molecular medicine, they underscore the molecular features that distinguish two groups of patients (e.g., responsive to therapy vs unresponsive).

The significance of a pattern is commonly assessed through *statistical hypothesis testing*: a statistical test is used to obtain a p -value that quantifies the probability that the association observed in the data is due to chance. The most commonly used test to assess the association of a pattern with class labels is Fisher's exact test [8]. Fisher's test is a *conditional test*: it assumes that the data generating process only produces datasets in which both the number of transactions in each class *and* the number of transactions in which the pattern appears are the same as in the observed dataset, i.e., it *conditions* on the observed variables of interest.

In contrast, *unconditional tests* such as Barnard's exact test [2] assumes that the frequency of the pattern observed in the real dataset is (the realization of) a *random* variable. Unconditional tests therefore assess the association between a pattern and class labels considering also scenarios (i.e., datasets) where the frequency of the pattern is different from what is observed in the real data. These tests seem therefore more appropriate for the significant pattern mining setting. The computation of the p -value for unconditional tests is usually more expensive than for conditional tests, since one needs to explore the space of the possible values for *nuisance parameters* describing the (unknown) values of the underlying process that generated the data. In significant pattern mining, the nuisance parameter of a pattern is the probability that it appears in a transaction generated by the underlying process.

? QUESTION FROM MATTEO (2/14): Part of the following paragraph(s) feels in part the same as in the ones above. Can we merge it better?

Conditional tests and unconditional tests are based on different assumptions regarding how data is generated and collected, namely, whether the variables of interests (e.g., patterns frequencies) would be the same in a different repetition of the experiment (*conditional tests*) or not (*unconditional tests*). To understand when the two situations arise, consider for example the study of the appearance of (behavioral) patterns (e.g., posting information regarding a specific topic) in members of two online communities (defining the two classes). When deciding to collect the data (e.g., whether a user

posted information on the topic or not), you may decide to stop once enough members (overall) have posted about the topic. In this case the assumption of conditional tests are met, since every repetition of the experiment would result in the exact same number of appearances of the pattern. In a different situation one may instead decide to collect data for a fixed amount of time: the number of times the pattern appears is, thus, not fixed, and would change among repetitions of the experiment. In this scenario, unconditional tests better reflect the process with which data is generated and collected.

In data mining, the latter scenario is far more common and natural than the former: data is collected from two different groups or conditions for some amount of time, and then the data is analyzed. However, conditional tests such as Fisher's are commonly employed in such scenarios. The difference between using conditional and unconditional tests is usually small when testing the significance of a *single* pattern S : in this case one can flag S as significant if its p -value is below a fixed threshold α with the guarantee that this corresponds to a *false discovery* (i.e., reporting S as significant when it is not) is bounded by α .

NOTE FROM MATTEO (2/18): There is a disconnect between the paragraph above and the one below. First we are saying that the difference between conditional and unconditional is small when testing only one pattern. Thus, one would expect that then we say that the difference is large when testing multiple patterns, but that punchline is never really delivered. I believe the issue is that we are deviating a bit by mentioning the challenges in multiple hypothesis testing and the possible solutions with too many details. Also, wouldn't be sufficient to say that for small p -values the difference between conditional and unconditional grows?

The situation is dramatically different in significant pattern mining, where a huge number of patterns appearing in the datasets are tested, resulting in a *multiple hypothesis testing* problem. In this case, if one tests m patterns, the expected number of patterns with p -values below α is $m\alpha$ even when no pattern is significant.

Several methods have been proposed to deal with multiple hypothesis testing [3, 5, 30]. These tests provide various guarantees, among which the most commonly considered are guarantees on the probability of one or more false discoveries, called Family-Wise Error Rate (FWER), and they all require a pattern to have very small p -value in order to be flagged as significant. For patterns with very small p -values, conditional tests and unconditional tests display strikingly different p -values (Fig. 1). In particular, Fisher's exact test produces p -values much below the p -values that are to be expected under the uniform distribution of p -values when there is no association between patterns and class labels, that may results in a much higher FWER than expected.

To-Do: Better explain the experiment, and what we are showing and how to interpret it. Also, what we say doesn't feel strong enough

To the best of our knowledge, no practical method to identify significant patterns with unconditional testing exists.

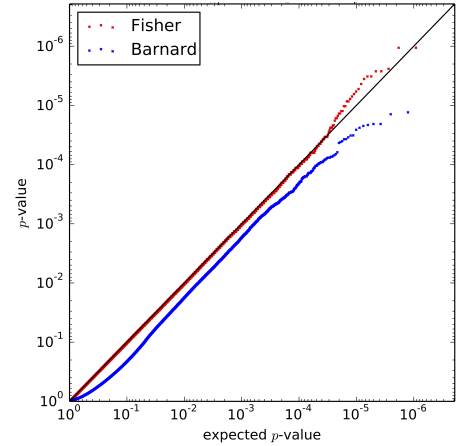


Figure 1: Observed vs. expected p -value for Fisher's and Barnard's exact tests. p -values for 10^6 random contingency tables for pattern S , with $n = 10^4$, $n_1 = 0.4 \cdot n$, n_0 to $n - n_1$, and $\sigma_0(S)$ generated from $\text{Bin}(n_0, 0.5)$ and $\sigma_1(S)$ from $\text{Bin}(n_1, 0.5)$ (i.e., sampling from the unconditional null distribution). See Sect. 3 for parameters' definitions. The p -values from Fisher's exact test are *smaller* than the expected p -values (the values on the axes *decrease* toward the right and upwards) and may lead to a *higher* FWER than expected when using multiple hypothesis testing corrections, and thus to false discoveries.

Contributions. We present SPuMANTE, the first efficient algorithm for mining significant patterns without conditioning on the observed values of the pattern frequencies and while controlling the FWER. In detail, our contributions are the following.

- At the core of SPuMANTE is UT, our novel unconditional statistical test for the significance of a single pattern. UT, being unconditional, is more appropriate for significant pattern mining. UT is, to our knowledge, the first computationally efficient unconditional test. To achieve this efficiency, it combines confidence intervals on the expected frequency of the pattern, with deep insights on the computation of bounds on the p -value, and a smart strategy to explore the space of contingency tables. UT's usefulness extends beyond its employment in SPuMANTE, and we are studying further applications of this novel test.
- SPuMANTE controls the FWER at level α , for an user-specified $\alpha \in (0, 1)$. To achieve this goal, we develop an efficient way to compute a lower bound to the p -value for UT, and use it by adapting the strategy used in LAMP [25]. SPuMANTE uses UT in combination with recently developed bounds on the maximum deviation of the observed frequency of a pattern from its expectation that hold simultaneously over all patterns [20], rather than having to expensively compute a different confidence interval for each pattern. To the best of our knowledge SPuMANTE is the first statistical test in which such uniform bounds have been used, and we believe that this approach could be applied to other algorithms for significant pattern mining.

- We evaluate SPuMANTE on real datasets and compare its performance with the state-of-the-art method LAMP [25], based on Fisher's exact test. The results show that SPuMANTE has high statistical power and is faster than LAMP in particular for large datasets, due to a high number of patterns that do not require the explicit computation of the p -value but can be flagged as significant based on the confidence intervals alone.

✓ **To-Do:** Mention some numbers?

2 RELATED WORK

Our work focuses on *significant* pattern mining [10]. Our goals is to develop significant pattern mining algorithm (called SPuMANTE) that (in no particular order) 1. controls the FWER; 2. uses a novel unconditional test; and 3. exhibits high statistical power.

Controlling the FWER can be done with classical statistical tools such as the Bonferroni's method, or its refinements [12]. These approach has been proposed in many previous works on significant pattern mining [27–29]. These methods exhibit limited statistical power, because of the very large number of associations being tested. For this reason, SPuMANTE does not use classical statistical approaches for the control of the FWER. Instead, SPuMANTE uses a strategy similar to the one proposed in LAMP [17, 25], a recently-introduced method for achieving higher statistical power while controlling the FWER. LAMP is designed for the Fisher's exact test, which is a *conditional*. SPuMANTE uses a significance-level-search strategy similar to the one in LAMP, profoundly modified to be appropriate for our novel test \mathcal{U} , which is *unconditional* and, as argued in Sect. 1, more appropriate for the significant pattern mining setting. Other works [15, 19, 24] use the Westfall-Young permutation test [30], that is more powerful than LAMP but also more computationally expensive. Our novel unconditional test \mathcal{U} can be used in combination with the Westfall-Young permutation testing as well.

Barnard's test [2] is an unconditional exact test to assess the statistical significance of associations from a 2x2 contingency table. Compared to Fisher's test [8], it offers more statistical power [4, 6, 16], and it only requires conditioning on one set of marginals of the conditionals tables, rather than two as Fisher's. The popularity of Barnard's test was hindered partially by Fisher's criticism, and partially by the excessive computational cost that a naïve implementation of the test requires. We do not enter the debate on which of the two tests to use [7], but our results suggest that unconditional tests like Barnard's exact test may be more appropriate for significant pattern mining. Rather, our work focuses on the computational aspects, and specifically in how to speed up the execution of the test. Our work is similar, in spirit, to that of Hämmäläinen [9], who studied computationally efficient upper bounds to the p -value of Fisher's test. Differences between the two statistical tests may become significant for small p -values (see Fig. 1), and thus become important in the multiple hypothesis testing scenario where they may lead to a lack of control on the FWER. Vandin et al. [26] observed such lack of control for other tests.

Variables	$S \subseteq t$	$S \not\subseteq t$	Row totals
$t \in \mathcal{Z}^1$	$\sigma_1(S)$	$n_1 - \sigma_1(S)$	n_1
$t \in \mathcal{Z}^0$	$\sigma_0(S)$	$n_0 - \sigma_0(S)$	n_0
Column totals	$\sigma(S)$	$n - \sigma(S)$	n

Table 1: 2×2 contingency table summarizing the appearance of the pattern S in \mathcal{Z}^0 and \mathcal{Z}^1 .

? **QUESTION FROM MATTEO (2/16):** Can we name names, instead of saying “other tests”?

✉ **NOTE FROM MATTEO:** Never use “this” alone, as in the following sentence. It took me a while to understand that we are saying that we are the first to report the discrepancy. In any case, I suggest to move the following sentence to the introduction where Fig. 1 is discussed, or rephrase it to refer to that figure.

To the best of our knowledge this is the first time that the observation is reported for Fisher's exact test.

We study the fundamental task of significant pattern mining, but extensions to this settings are possible and have been studied. Komiyama et al. [13] look at emerging patterns that significantly differ between two databases. Papaxanthos et al. [18] and Terada et al. [23] present methods to mine significant patterns in the presence of confounding variables and covariates, while Sugiyama et al. [21] focuses on mining significant subgraphs from networks, and He et al. [11] look at the extraction of significant sequential patterns. We believe our method can be extended to these other settings and we will study these extensions in future works.

3 PRELIMINARIES

Let \mathcal{I} be an alphabet of ordered *items*, and let $\{\ell^0, \ell^1\}$ be two (*class*) *labels*, i.e., items that do not belong to \mathcal{I} . A *dataset*

$$\mathcal{D} = \{(t_1, \ell_1), (t_2, \ell_2), \dots, (t_n, \ell_n)\}$$

is a multiset of $|\mathcal{D}| = n$ pairs (t_i, ℓ_i) where $t_i \subseteq \mathcal{I}$ is a *transaction*, and $\ell_i \in \{\ell^0, \ell^1\}$ is a label, for $1 \leq i \leq n$. The multiset of the first elements of the pairs in \mathcal{D} is naturally partitioned into two multisets \mathcal{Z}^0 and \mathcal{Z}^1 , where \mathcal{Z}^i contains all and only the first elements of the pairs in \mathcal{D} with second element ℓ^i , for $i \in \{0, 1\}$. We define $n_i = |\mathcal{Z}^i|$, with $n_1 + n_0 = n$.

A *pattern* (or itemset) S is a set of items, $S \subseteq \mathcal{I}$. We say that S *appears* in a transaction t if $S \subseteq t$, and say that t *contains* S . The *support* $\sigma_i(S)$ (resp. *frequency* $f_i(S)$) of S in \mathcal{Z}^i is the *number* (resp. *fraction*) of transactions in \mathcal{Z}^i that contain S , for $i \in \{0, 1\}$. With a slight abuse of notation, we denote as $\sigma(S)$ (resp. $f(S)$) the number (resp. fraction) of transactions in the pairs of \mathcal{D} that contain S . Thus, $f_i(S) = \sigma_i(S)/n_i$, $i \in \{0, 1\}$, and $f(S) = \sigma(S)/n$. For any pattern S , these quantities (and their complements) are summarized in a 2×2 contingency table such as the one in Table 1.¹

In significant pattern mining, the dataset \mathcal{D} is assumed to be the product of a stochastic process that generates sets of pairs

¹The “2×2” attribute refers to the four central cells of the table, which contain the actual information about the pattern S .

(t, λ). Different assumptions can be made on this process (details in Sect. 3.1 and 3.2). Independently on the assumptions, for any pattern $S \subseteq \mathcal{I}$, we let $\pi_{S,i}$ be the probability that a pair (t, λ) generated by the process is such that $S \subseteq t$ and $\lambda = \ell^i$, for $i \in \{0, 1\}$.

The key task in significant pattern mining is to identify the patterns that exhibit a *significant association* with one of the two labels, i.e., for which $\pi_{S,0} \neq \pi_{S,1}$. Given a single pattern S , assessing the statistical significance of an association corresponds to using the *observed* contingency table for S to evaluate whether it *supports* the *null hypothesis* $H_S : \pi_{S,0} = \pi_{S,1}$, i.e., whether the observed data \mathcal{D} could have been generated from a process satisfying H_S .

The only available information about S are contained in the contingency table, thus we cannot be *deterministically certain* in our assessment of the significance of the association of S with one label: due to the randomness involved in the data generation process, there is the possibility of flagging the association as significant when it is not, i.e., of making a *false discovery*.

The assessment of whether the null hypothesis for S is supported by the observed contingency table for S involves computing a *p-value* p_S . This quantity is defined as the probability, w.r.t. the data generating distribution and under the assumption that the null hypothesis is true, of observing a contingency table for S that is *as or more extreme* (i.e., has equal or lower probability of being observed) than the one that is actually observed. The set of more extreme contingency tables and the probability associated to each of them depend on the *conditions* imposed on the data generation process (details in Sect. 3.1 and 3.2).

No matter what the conditions are, once the value of p_S or an *upper bound* to it is known, *rejecting* the null hypothesis H_S , i.e., flagging the pattern S as having a significant association with one of the two labels *iff* $p_S \leq \delta$ ensure that the probability (w.r.t. the randomness in the generating process) that S is a false discovery is not greater than $\delta \in (0, 1)$.

3.1 Conditional testing

Let's focus for now on a single pattern $S \subseteq \mathcal{I}$. A set of conditions commonly imposed on the data generating process, made for example by the famous Fisher's exact test [8] is to *condition on all values in the marginals* (i.e., in the bottom row and rightmost column) of the observed contingency table for S : the space of possible contingency tables for S contains all and only those with the same values for n , n_1 (thus n_0), and $\sigma(S)$, as in the observed one.

Under these conditions and assuming that the null hypothesis H_S is true, the quantity $\sigma_1(S)$ follows a *hypergeometric distribution*: given $a \in [0, \min\{n_1, \sigma(S)\}]$, the probability $P^{(F)}(a)$ (the "F" stands for "Fisher") of observing a contingency table for S where $\sigma_1(S) = a$ is

$$P^{(F)}(a) = \frac{\binom{n_1}{a} \binom{n_0}{\sigma(S)-a}}{\binom{n}{\sigma(S)}} .$$

Let b be the value for $\sigma_1(S)$ in the observed contingency table for S , the *p-value* $p_S^{(F)}$ is then

$$p_S^{(F)}(b) = \sum_{a : P^{(F)}(a) \leq P^{(F)}(b)} P^{(F)}(a) . \quad (1)$$

Drawbacks. Imposing on the generative process the conditions we just described may be reasonable when only considering a single pattern S . In the significant pattern mining setting, the space of possible patterns is the powerset of \mathcal{I} . Since there is a single generative process, one would have to impose on it that, for *each* S of the $2^{|\mathcal{I}|}$ possible patterns, the generative process only generates contingency tables for S with the observed value of $\sigma(S)$. Imposing such a large number of conditions seems excessively restrictive.

3.2 Unconditional testing

A more reasonable set of conditions is to consider only n and n_1 (and thus n_0) fixed as in the observed dataset. These conditions depend only on the observed dataset, and not in any way on the observed frequencies (in \mathcal{D}) of any of the patterns. With a somewhat unfortunate name choice, tests that impose such conditions, e.g., Barnard's (exact) test [2], are known as *unconditional tests*. For a pattern S , the space of possible contingency tables contains all and only those with the same values for n and n_1 (thus n_0), as in the observed one. The value for $\sigma(S)$ is not fixed, hence not known a priori. This set of assumptions is much more reasonable for the significant pattern mining setting, where the value of $\sigma(S)$ for each pattern S is unknown and must be obtained by running a pattern mining algorithm on the dataset.

To define the *p-value* for S we need to first introduce the concept of the *nuisance parameter* $\pi \in [0, 1]$. The nuisance parameter is the *assumed value* for $\pi_{S,0}$ and $\pi_{S,1}$ under the null hypothesis that these quantities are equal.

Under the above conditions, and assuming that the null hypothesis H_S is true, and assuming a fixed, known value for the nuisance parameter π , the probability of observing a contingency table for S , with $\sigma(S) = x$ and $\sigma_1(S) = a$, for $x \in [0, n]$, and $a \in [0, \min\{x, n_1\}]$, is

$$P(x, a \mid \pi) = \binom{n_0}{x-a} \binom{n_1}{a} (\pi)^x (1-\pi)^{n-x} .$$

The *conditional p-value* $p_S(\pi)$ is then the sum of the probabilities of possible contingency tables with probability no greater than the probability for the observed contingency table of S :

$$p_S^{(B)}(\pi) = \sum_{(x,a) : P(x,a \mid \pi) \leq P(\sigma(S), \sigma_1(S) \mid \pi)} P(x, a \mid \pi) .$$

To obtain the actual *p-value* for S or an upper bound to it which, we recall, is sufficient to perform the test, it is necessary to eliminate the dependency on nuisance parameter π . For example, Barnard's test [2] uses as upper bound $p_S^{(B)}$ the maximum of $p_S^{(B)}(\pi)$ over all values of $\pi \in (0, 1)$:

$$p_S^{(B)} = \max \left\{ p_S^{(B)}(\pi), 0 \leq \pi \leq 1 \right\} .$$

Finding this maximum is computationally expensive. One of our goals is designing an unconditional test with an upper bound to the *p-value* that is efficient to compute (see Sect. 4.1).

3.3 Multiple hypothesis testing and LAMP

When a single pattern S is tested, flagging it as *significant* when its *p-value* is smaller than a *significance threshold* $\delta \in [0, 1]$, fixed *a priori*, guarantees that the probability of a false discovery (i.e., reporting S as significant when it is not) is bounded by δ . If such

approach is followed when testing $h > 1$ patterns (i.e., *multiple hypotheses*, the expected number of false discoveries is

? QUESTION FROM MATTEO (2/16): “could be as large as”?

δh . As h grows large, as is typically the case for significant pattern mining, the probability of having at least one (and possibly many) false discovery also grows. An appropriate *multiple hypothesis testing correction* of the significance threshold used for each test is needed to ensure a low probability of false discoveries.

One common approach is to perform such a correction to bound the *Family-Wise Error Rate* FWER, i.e., the probability of reporting a *at least one* false discovery, where the sample space is the set of possible datasets generated by the underlying process.

For a given $\delta \in [0, 1]$, using δ as the significance threshold for each test, i.e., rejecting all null hypotheses with p -value $\leq \delta$ would lead to a FWER FWER(δ). In most cases it is not possible to derive a closed formula for FWER; a more tractable and desirable inverse approach allows to obtain, given an acceptable-FWER threshold $\alpha \in (0, 1]$, a so-called *corrected significance threshold* δ to use in each test in order to have $\text{FWER}(\delta) \leq \alpha$. A simple, but conservative, way to set δ is the Bonferroni’s correction [5]: given α , use $\delta = \alpha/h$ to guarantee that $\text{FWER}(\alpha) \leq h\delta = \alpha$. The problem with Bonferroni’s correction is that, for large h , δ is close to 0, resulting in low *statistical power*, i.e., many actually significant patterns are not flagged as such.

? QUESTION FROM MATTEO (2/16): Shall we cite Webb’s works? [27–29]

More sophisticated techniques have been devised to obtain higher statistical power while keeping the FWER at most α . Tarone [22] introduces the notion of *minimum attainable (min. att.) p -value*: if we use a corrected significance threshold δ , patterns whose min. att. p -value is greater than δ , called *untestable*, do not need to be counted among the h to be tested, thus should not be considered in the computation of δ with the Bonferroni’s correction. Defining $k(\delta)$ as the number of patterns with min. att. p -value $\leq \delta$, the corrected significance threshold can be obtained as $\delta = \alpha/k(\delta)$.

Terada et al. [25] present LAMP, a significant pattern mining algorithm that uses the concept of min. att. p -value for Fisher’s exact test: for a given pattern S , its p -value p_S can be expressed as a function of $\sigma_1(S)$ only (see (1)). Since the set of possible values of $\sigma_1(S)$ is $[0, \min\{\sigma(S), n_1\}]$, the min. att. p -value $\psi(S)$ is

$$\psi^{(L)}(S) = \min \left\{ p_S^{(F)}(a), a \in [0, \min\{\sigma(S), n_1\}] \right\}.$$

For a candidate corrected significance threshold δ , if $\psi(S) > \delta$ the pattern S will never be flagged as significant and is therefore *untestable*. In order to be used in a significant pattern mining setting, Terada et al. [25] introduce an easy-to-compute monotonically decreasing lower bound $\check{\psi}(S)$ to $\psi(S)$ (assuming w.l.o.g. that $n_1 \leq n_0$):

$$\check{\psi}^{(L)}(S) = \begin{cases} \psi^{(L)}(S) & \text{if } 0 \leq \sigma(S) \leq n_1 \\ 1/\binom{n}{n_1} & \text{if } n_1 < \sigma(S) \leq n \end{cases}.$$

✓ To-Do FROM MATTEO (2/14): It is never mentioned (and too unclear without mentioning it) what the above function is monotonically decreasing in.

SPuMANTE employs a strategy

✓ To-Do FROM MATTEO (2/16): We don’t say a strategy for what

similar to the one in LAMP, but uses a new unconditional test (see Sect. 4.1), thus requiring the development of an efficiently computable lower bound to the min. att. p -value for a pattern S using such test (Sect. 4.2.2).

4 SIGNIFICANT PATTERN MINING WITH UNCONDITIONAL TESTING

We now describe SPuMANTE, our algorithm for significant pattern mining with unconditional testing. We start by introducing UT, a novel Unconditional Test that is based on confidence intervals for the expected frequencies of the patterns. Due to space constraints, some of our proofs are presented in the Appendix.

4.1 The UT test

Let S be the pattern of which we are assessing the association with the labels. Our novel Unconditional Test (UT) assumes to know two confidence intervals $C_0(S)$ and $C_1(S)$, for $\pi_{S,0}$ and $\pi_{S,1}$, respectively, s.t. the event $E_S = “\pi_{S,0} \in C_0(S) \text{ and } \pi_{S,1} \in C_1(S)”$ holds with probability $1 - \gamma$ (over the randomness in the data generation process). We discuss in Sect. 4.2 how such confidence intervals can be obtained *simultaneously* for all patterns S . Let the interval $C(S)$ be $C(S) = C_0(S) \cap C_1(S)$.

To define the p -values used by UT, we need some preliminary definitions. For $y \in [0, n]$, $b \in [0, \min\{y, n_1\}]$, and $\pi \in (0, 1)$, define $T(y, b, \pi)$ as the set of pairs (x, a) such that $P(x, a | \pi) \leq P(y, b | \pi)$ and define the function

$$\phi(y, b, \pi) = \sum_{(x, a) \in T(y, b, \pi)} P(x, a | \pi).$$

The value $\phi(y, b, \pi)$ is the probability, under the null hypothesis $\pi_{S,0} = \pi_{S,1}$ and for a fixed value π of the nuisance parameter, to observe a contingency table as or more extreme than the one with $\sigma(S) = y$, $\sigma_1(S) = b$, and $\sigma_0(S) = y - b$ (see Table 1). We define the p -value p_S conditioned on the event E_S above as

$$p_S = \begin{cases} 0 & \text{if } C(S) = \emptyset \\ \max\{\phi(\sigma(S), \sigma_1(S), \pi), \pi \in C(S)\} & \text{othw.} \end{cases}.$$

The p -value p_S is a *conditional probability*: it is the probability of observing contingency tables at least as extreme as the one seen in the dataset *conditioning* on the event E_S .

Given a fixed threshold α , UT flags S as significant *iff* $p_S \leq \alpha - \gamma$. The following property holds.

THEOREM 4.1. *Let S be a fixed pattern. The probability that UT flags S as significant when it is not is $\leq \alpha$.*

PROOF. Let F be the event “UT flags S as significant when it is not”, which corresponds to a false discovery. It holds

$$\Pr(F) = \Pr(F | E_S) \Pr(E_S) + \Pr(F | \bar{E}_S) \Pr(\bar{E}_S) \leq \Pr(F | E_S) + \Pr(\bar{E}_S),$$

where \bar{E}_S denotes the event complementary to E_S . By the definition of confidence interval, $\Pr(\bar{E}_S) \leq \gamma$, while $\Pr(F | E_S) \leq \alpha - \gamma$ since when E_S holds we are using the standard hypothesis testing framework with significance threshold $\alpha - \gamma$ and the p -value p_S . \square

4.1.1 An upper bound to the p -value. The exact computation of p_S when $C(S) \neq \emptyset$ requires an expensive search over the values of $\pi \in C(S)$. For the purpose of testing the significance of a pattern and ensuring that Thm. 4.1 still holds, only an efficient-to-compute *upper bound* to the p -value is needed. We prove the following.

THEOREM 4.2. $p_S \leq P(\sigma(S), \sigma_1(S) \mid f(S))(n_0 + 1)(n_1 + 1)$.

The upper bound $\hat{p}_S = P(\sigma(S), \sigma_1(S) \mid f(S))(n_0 + 1)(n_1 + 1)$ can be computed in $O(1)$ time.

4.1.2 A lower bound to the p -value. We now show a lower bound to p_S . More than being just of theoretical interest, this lower bound is the starting point to derive, in Sect. 4.2.2, an efficiently-computable, monotonically-decreasing lower bound to the minimum attainable p -value for a pattern S (required to compute the corrected significance threshold in a way similar to what is done by LAMP [25]).

Computing our lower bound does not require an expensive search over the values of π , thanks to the following result.

LEMMA 4.3. If $C(S) \neq \emptyset$, then $p_S \geq \phi(y, b, \pi)$ for any $\pi \in C(S)$.

PROOF. Follows from the definition of p_S , that is the maximum over $\pi \in C(S)$ of the r.h.s. \square

Lemma 4.3 states that any $\pi \in C(S)$ allows to obtain a lower bound to p_S , so we choose $\pi = f(S)$, and define the lower bound \check{p}_S to p_S as

$$\check{p}_S = \phi(\sigma(S), \sigma_1(S), f(S)) . \quad (2)$$

Our choice for π is driven by the objective of maximizing the number of contingency tables in $T(\sigma(S), \sigma_1(S), \pi)$, which we try heuristically to achieve by maximizing $P(\sigma(S), \sigma_1(S) \mid \pi)$, which is straightforward as shown in the following result.

PROPOSITION 4.4. It holds $\arg \max_{\pi} \{P(x, a \mid \pi)\} = x/n$.

We show in Sect. 5.1 that \check{p}_S provides a tight lower bound to p_S .

4.2 SPUMANTE: mining significant patterns

We now describe SPUMANTE, our algorithm for mining significant patterns with UT while controlling the FWER. First, we need to discuss some important technical facts.

4.2.1 Simultaneous confidence intervals. For each tested pattern S , UT requires confidence intervals for $\pi_{S,0}$ and $\pi_{S,1}$. Rather than computing these confidence intervals separately for each pattern, SPUMANTE uses recently developed probabilistic bounds to the maximum deviation of the frequency of a pattern from its expectation [20] to derive confidence intervals for the quantities $\pi_{S,0}$ and $\pi_{S,1}$ of every pattern that hold *simultaneously* with high probability. Specifically, given \mathcal{D} and a confidence parameter $\gamma \in (0, 1)$, we use a modified version of the work by Riondato and Upfal [20], called AMIRA, to obtain a value $\varepsilon \in (0, 1)$ with the following property. Given this ε , define, for each pattern S , the intervals $C_0(S)$ and $C_1(S)$ as

$$C_0(S) := \left[f_0(S) - \varepsilon \frac{n}{n_0}, f_0(S) + \varepsilon \frac{n}{n_0} \right],$$

$$C_1(S) := \left[f_1(S) - \varepsilon \frac{n}{n_1}, f_1(S) + \varepsilon \frac{n}{n_1} \right],$$

and define the event $E_{S,\varepsilon} = \pi_{S,0} \in C_0(S) \text{ and } \pi_{S,1} \in C_1(S)$. Consider the event $E_\varepsilon = \bigcap_{S \subseteq \mathcal{I}} E_{S,\varepsilon}$. The ε returned by AMIRA when run on \mathcal{D} with confidence parameter γ is such that E_ε holds with probability at least $1 - \gamma$ (w.r.t. the randomness in the data generative process).

4.2.2 Lower bound to the min. att. p -value. In order to use UT in our algorithm SPUMANTE to efficiently mine significant patterns while rigorously controlling the FWER, we need to define a *monotone lower bound* $\check{\psi}(x)$ to the minimum attainable p -value of any S for which $\sigma(S) \leq x$, $x \in [0, n]$. Having this lower bound is crucial to prune the search space of testable patterns.

Our bounds crucially hinges on the following result.

THEOREM 4.5. Let $C_0(S) \cap C_1(S) = C(S) \neq \emptyset$. Then $f(S) \in C(S)$.

It is easy to show that

? QUESTION FROM MATTEO (2/15): Is it really easy to show it?

under the ongoing assumption $n_1 \leq n_0$ it holds

$$\arg \min_a \{\phi(x, a, \pi)\} = \min\{x, n_1\} \quad (3)$$

and that, for $x \leq n_1$, it holds that

$$\begin{aligned} \min \left\{ \phi \left(y, b, \frac{y}{n} \right), y \leq x, 0 \leq b \leq y \right\} &= \\ &= \min \left\{ \phi \left(x, b, \frac{x}{n} \right), 0 \leq b \leq x \right\} \\ &= \phi \left(x, x, \frac{x}{n} \right), \end{aligned}$$

where we used (3) in the last step. Then, for $x \in [0, n]$ let

$$Q(x) = \left\{ (r, w), r \in [0, n_0], w \in [0, n_1] \text{ s.t. } r + w \leq x \right. \\ \left. \wedge \left(\frac{r}{n_0} + \varepsilon \frac{n}{n_0} < \frac{w}{n_1} - \varepsilon \frac{n}{n_1} \vee \frac{w}{n_1} + \varepsilon \frac{n}{n_1} < \frac{r}{n_0} - \varepsilon \frac{n}{n_0} \right) \right\}$$

and let

$$\psi(x) = \begin{cases} 0 & \text{if } Q(x) \neq \emptyset \\ \phi(x, x, x/n) & \text{otherwise} \end{cases} .$$

Checking whether $Q(x)$ is non-empty can be done by checking if it contains either $(0, x)$ or $(x, 0)$: if $Q(x)$ does not contain either, it is easy to show that it must be empty.

? QUESTION FROM MATTEO (2/15): Is it really easy to show it?

Define

$$\check{\psi}(x) = \begin{cases} \psi(x) & \text{if } x < n_1 \\ \psi(n_1) & \text{if } x \geq n_1 \end{cases} . \quad (4)$$

$\check{\psi}(x)$ is our lower bound to the minimum attainable p -value. SPUMANTE uses it to heck whether to mark a pattern S with $\sigma(S) = x$ as untestable when looking for the corrected significance threshold δ (see Sect. 4.2).

? QUESTION FROM MATTEO (2/15): Where do we discuss that $\check{\psi}$ is monotonically decreasing (in x)? We do not even mention this fact.

Efficient computation of ϕ . After having defined \tilde{p}_S and $\tilde{\psi}(x)$, we still have to address how to compute them efficiently in the case $C(S) \neq \emptyset$, i.e., how to compute the value $\phi(y, b, \pi)$ efficiently. A naïve approach is to enumerate *all* $(x, a_1) \in T(y, b, \pi)$; since this set is not known a priori (i.e., there is no simple algorithm to generate *only* pairs (x, a_1) that are in $T(y, b, \pi)$), this approach requires computing $P(a_0 + a_1, a_1 \mid \pi)$ for all possible pairs $(a_0, a_1) \in [0, n_0] \times [0, n_1]$, leading to the computation of $\Theta(n_0 n_1)$ probabilities. As we discuss in Sect. 5.1, even for samples of moderate size this approach is not feasible in reasonable time.

✓ **TO-DO FROM MATTEO (2/16):** The above issue is actually mentioned (I wouldn't say "discussed") just before Sect. 5.1.

Enumerating only pairs $(x, a_1) \in T(y, b, \pi)$ or only pairs $(x, a_1) \notin T(y, b, \pi)$ would still require to evaluate $P(x, a_1 \mid \pi)$ a corresponding number of times, i.e., in the order of $\Theta(\min\{|T(y, b, \pi)|, n_0 n_1 - |T(y, b, \pi)|\})$, which is impractical for most cases. We address this issue with an efficient algorithm to compute $\phi(y, b, \pi)$ while avoiding the enumeration of many contingency tables, thanks to the novel formulation of $\phi(y, b, \pi)$ provided by the following result.

PROPOSITION 4.6. *Let $y \in [0, n]$, $b \in [0, \min\{y, n_1\}]$, and $\pi \in (0, 1)$. Define $A_1 = \{a_1 : P(a_1 + \lfloor (n_0 + 1)\pi \rfloor, a_1 \mid \pi) > P(y, b \mid \pi)\}$, and define the set $A_{0, a_1} = \{a_0 : P(a_1 + a_0, a_1 \mid \pi) \leq P(y, b \mid \pi)\}$. Then*

$$\begin{aligned} & \sum_{(x, a) \in T(y, b, \pi)} P(x, a \mid \pi) \\ &= \sum_{a_1 \notin A_1} B(a_1, n_1, \pi) + \sum_{a_1 \in A_1} \left(B(a_1, n_1, \pi) \sum_{a_0 \in A_{0, a_1}} B(a_0, n_0, \pi) \right), \end{aligned} \quad (5)$$

where $B(z, h, \pi) = \binom{h}{z} \pi^z (1 - \pi)^{h-z}$ is the probability of obtaining z successes on h independent trials with success probability π .

This formulation leads to the efficient algorithm to compute $\phi(y, b, \pi)$ shown in Alg. 1, where we use the *incomplete beta function* to compute the cumulative distribution function (CDF) for Binomial distributions.

In fact, if we let $F(a, n, \pi) = \sum_{a'=0}^a B(a', n, \pi)$ be the CDF for value a of the Binomial distribution of parameters n, π , we can compute the terms of (5) with $O(|A_1|)$ computations (Alg. 1, line 7) of the incomplete beta function $\beta_{1-\pi}(n+1-a, a+1) = F(a, n, \pi)$ that is efficiently computable using Lentz's algorithm [14], a fast and precise method to evaluate continued fractions. The computation of values a' and a'' (lines 6–7) across *all* iterations of the **forall** loop requires two binary searches for the first iteration, while their values in subsequent iterations can be computed with $O(1)$ operations by updating their previously computed values. Therefore, lines 6–7 require $O(\log(n_0))$ operations across *all* iterations of the **forall** loop. (All details will be given in the full version of this work.)

? **QUESTION FROM MATTEO (2/15):** Why not in the appendix?

Algorithm 1: Efficient computation of $\phi(y, b, \pi)$

Input: $y \in [0, 1]$, $b \in [0, \min\{n_1, y\}]$, $\pi \in [0, 1]$
Output: $\phi(y, b, \pi)$.

```

1  $v \leftarrow 0$ 
2  $z \leftarrow P(y, b \mid \pi)$ 
3  $a'_0 \leftarrow \lfloor (n_0 + 1)\pi \rfloor$ 
4  $A_1 \leftarrow \{a_1 : P(a_1 + a'_0, a_1 \mid \pi) > z\}$ 
5 forall  $a_1 \in A_1$  do
6    $a' \leftarrow \min_{a_0} \{a_0 \leq a'_0 \mid P(a_0 + a_1, a_1 \mid \pi) > z\}$ 
7    $a'' \leftarrow \min_{a_0} \{a_0 > a'_0 \mid P(a_0 + a_1, a_1 \mid \pi) \leq z\}$ 
8    $p' \leftarrow \binom{n_1}{a_1} \pi^{a_1} (1 - \pi)^{n_1 - a_1}$ 
9    $v \leftarrow v + \beta_\pi(a', n_0 - a') p' + \beta_{1-\pi}(n_0 + 1 - a'', a'') p'$ 
10  $a' \leftarrow \min_{a_1} \{a_1 > \max\{A_1\}\}$ 
11  $a'' \leftarrow \max_{a_1} \{a_1 < \min\{A_1\}\}$ 
12  $v \leftarrow v + \beta_\pi(a' + 1, n_0 + 1 - a') + \beta_{1-\pi}(n_0 + 1 - a'', a'')$ 
13 return  $v$ 
```

✓ **TO-DO FROM LEONARDO (3/13):** Update the algorithm (or maybe only explain in the description) with the following idea. To compute (5) we only need 4 calls in total to the incomplete beta function: 2 at the end, and 2 at the first iteration of the **forall** loop. In the other iterations of the **forall** loop I can update v with $O(1)$ operations, since I only have to update the quantities from previous iterations.

SPuMANTE outputs a set of significant patterns in \mathcal{D} with FWER $\leq \alpha$. Its pseudocode is presented in Alg. 2. SPuMANTE first obtains (line 1) the maximum deviation ε from AMIRA with parameter γ , so that the event E_ε holds with probability $\geq 1 - \gamma$. Then (line 2), SPuMANTE uses the lower bound $\tilde{\psi}(x)$ derived in Sect. 4.1.2 (and computed using Alg. 1) together with a strategy similar to the one in LAMP [25] to efficiently derive a corrected significance threshold δ to use in each test while ensuring that the FWER is at most $\alpha - \gamma$. SPuMANTE then loops over the items in \mathcal{I} and for each i of them, if $\tilde{\psi}(\sigma(\{i\})) \leq \delta$ (i.e., if $\{i\}$ is *testable*), $\{i\}$ is processed using the processPattern procedure. This procedure first computes the interval $C(S)$ (lines 6–8), and then computes the upper bound \hat{p}_S to the p -value (lines 9–12). If $C(S) = \emptyset$, \hat{p}_S is set to 0 (line 9); otherwise SPuMANTE computes \hat{p}_S using the bound from Thm. 4.2. SPuMANTE uses the upper bound \hat{p}_S to decide whether S is significant, returning S in output if $\hat{p}_S < \delta$ (line 13). Then (lines 14–15), the current pattern S is “grown” adding appropriate items from \mathcal{I} , enumerating the space of the testable patterns exhaustively in a depth-first order.

We can show the following property of SPuMANTE.

THEOREM 4.7. *The output of SPuMANTE has FWER $\leq \alpha$.*

PROOF (SKETCH). Consider the event F = “the number of false discoveries reported by SPuMANTE is > 0 ”. The FWER of the output of SPuMANTE is $\Pr(F)$. Recalling the event E_ε defined in Sect. 4.2, let \bar{E}_ε be the complementary event. It holds:

$$\Pr(F) = \Pr(F \mid E_\varepsilon) \Pr(E_\varepsilon) + \Pr(F \mid \bar{E}_\varepsilon) \Pr(\bar{E}_\varepsilon) \leq \Pr(F \mid E_\varepsilon) + \Pr(\bar{E}_\varepsilon).$$

Using AMIRA with parameter γ guarantees that $\Pr(\bar{E}_\epsilon) \leq \gamma$. By employing the LAMP strategy with parameter $\alpha - \gamma$ and using the upper bound \hat{p}_S to decide if S is significant, it holds that $\Pr(F | E_\epsilon) \leq \alpha - \gamma$.

✓ **To-Do from Matteo (2/15):** We really need to do something about this idea of the “strategy”: it is never explained and either a reader knows it or it is a black box and they have no clue of its properties, so mentioning it in the proof does not help them.

Therefore $\Pr(F) \leq \alpha - \gamma + \gamma = \alpha$. \square

4.2.3 Increasing the power of UT . SPuMANTE provides an efficient method to identify all significant patterns with bounded FWER. However, while extremely fast to compute, the upper bound of Thm. 4.2 does not always provide a tight approximation to the p -value of a pattern S , resulting in a potential reduction in power, even if as we show in Sect. 5.1 the most significant patterns are still reported. In the scenarios where one is interested in reporting a larger number of patterns, at the expense of weakening the guarantees of the FWER, one can use the lower bound \check{p}_S of Sect. 4.1.2 in place of the upper bound of Thm. 4.2 in lines 11–12. While in this case there is no guarantee on the FWER of the additional patterns reported (in addition to the ones obtained when the upper bound of Thm. 4.2 is used), in all our experiments (Sect. 5.1) we never observed false discoveries in the output, due to \check{p}_S being very close to the actual p -value p_S .

Algorithm 2: SPuMANTE

Input: Dataset \mathcal{D} , acceptable FWER $\alpha \in (0, 1)$, confidence parameter $\gamma \in (0, \alpha)$.

Output: Set of significant patterns with FWER $\leq \alpha$.

```

1  $\epsilon \leftarrow \text{AMIRA}(\mathcal{D}, \gamma)$ 
2  $\delta \leftarrow \text{correctedSignificanceThreshold}(\alpha - \gamma)$ 
3 forall  $i \in \mathcal{I}$  do
4   if  $\check{\psi}(\{i\}) \leq \delta$  then  $\text{processPattern}(\{i\})$ 
5 Function  $\text{processPattern}(S)$ 
6    $C_0(S) \leftarrow [f_0(S) - \epsilon \frac{n}{n_0}, f_0(S) + \epsilon \frac{n}{n_0}]$ 
7    $C_1(S) \leftarrow [f_1(S) - \epsilon \frac{n}{n_1}, f_1(S) + \epsilon \frac{n}{n_1}]$ 
8    $C_S \leftarrow C_0(S) \cap C_1(S)$ 
9   if  $C_S = \emptyset$  then  $\hat{p}_S \leftarrow 0$ 
10  else
11     $\hat{p}_{S,\pi} \leftarrow P(\sigma(S), \sigma_1(S) | f(S))$ 
12     $\hat{p}_S \leftarrow \hat{p}_{S,\pi}(n_0 + 1)(n_1 + 1)$ 
13  if  $\hat{p}_S \leq \delta$  then output  $S$ 
14  forall  $i \in \mathcal{I}$  s.t.  $i > \max\{j \in S\}$  do
15    if  $\check{\psi}(\sigma(S \cup \{i\})) \leq \delta$  then  $\text{processPattern}(S \cup \{i\})$ 
```

5 EXPERIMENTAL EVALUATION

We implemented SPuMANTE and tested it on several datasets. Our experimental evaluation has the following goals:

- assess the tightness of the lower bound \check{p}_S from (2) w.r.t. the exact p -value.

dataset	$ \mathcal{D} $	$ \mathcal{I} $	avg	n_1/n
svmguid3	1,243	44	21.9	0.23
mushroom	8,124	118	22	0.48
breast cancer	12,773	1,129	6.7	0.09
a9a	32,561	247	13.9	0.24
retail(U)	88,162	16,470	10.3	0.47
ijcnn1	91,701	44	13	0.10
cod-rna	271,617	16	8	0.33
covtype	581,012	64	11.9	0.49

Table 2: Datasets statistics. For each dataset we report: name (see Sect. 5 for the meaning of U), number $|\mathcal{D}|$ of transactions; the number $|\mathcal{I}|$ of items; average transaction length avg; fraction n_1/n of transactions in \mathcal{Z}^1 .

- evaluate the performance [✓ too vague] of UT : since no other method for performing efficiently an unconditional test for significant patterns exists, we compare UT with Fisher’s exact test, the de-facto standard *conditional* test employed for significant pattern mining algorithms.
- assess the effectiveness and the impact of the upper bound $\hat{\phi}$ and of the AMIRA confidence intervals on reporting significant patterns.

Implementation and environment. We implemented SPuMANTE² and UT by modifying a C implementation of LAMP³. For computing the incomplete beta function in lines 9 and 12 of Alg. 1, we use a publicly available implementation⁴ based on Lentz’s algorithm [14]. All the code was compiled with GCC 8 and run on a machine with a 16-core 2.30 GHz Intel Xeon CPU, 512 GB of RAM, on Ubuntu 14.04.

Datasets. We tested SPuMANTE on eight datasets commonly used for the benchmark of significant pattern mining algorithms and gathered from FIMI’04⁵, libSVM⁶, and UCI⁷. Descriptive statistics for these datasets are shown in Table 2. For datasets whose transactions are not naturally divided in two groups (marked with U), we selected the single item whose frequency is closer from below to 0.5, removed the corresponding item from every transaction, and use its appearance to divide the dataset in two groups. The reported ratio n_1/n refers to the output of this process. For real-valued features we obtained two bins by thresholding at the mean value and using one item for each bin.

Parameters and experiments. In all our experiments, we set $\alpha = 0.05$ and $\gamma = 0.01$. In order to study the impact of the dataset size on SPuMANTE’s performance, for all datasets we generate a random sample of size s by taking s transactions uniformly at random with replacement, varying s in the interval $[10^3, 10^6]$.

²The code of SPuMANTE and the scripts to replicate all experiments described in the paper are available at <https://goo.gl/sxx8JX>. See also the Appendix.

³<https://github.com/flinares/wylight>

⁴<https://github.com/codeplea/incbeta>

⁵<http://fimi.ua.ac.be>

⁶<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>

⁷<https://archive.ics.uci.edu/ml/index.php>

✓ **To-Do from MATTEO (2/15):** With reference to the notation introduced in the following paragraph: we cannot use UT or similar labels to denote variants of SPuMANTE. UT is our test. We need different notation (also the macros have the wrong names because they are not descriptive).

We enumerate the significant patterns using four different variants of SPuMANTE: the first, UT , is the one described in Sect. 4.2 that uses the confidence intervals of Sect. 4.2.1 and the upper bound from Thm. 4.2 to test the patterns; the second, that we denote UT^* , is the one described in Sect. 4.2.3, that provides increased power at the expense of relaxed guarantees on FWER; the third version, UT^C , flags an itemset S as significant only if its confidence interval $C(S)$ is $C(S) = \emptyset$; the last one, UT_n , is a naïve implementation of Alg. 1, that enumerates all the contingency tables for every pattern S , fixing π to $f(S)$. However, we do not include the results for UT_n , since its naïve enumeration strategy results in impractical running times: for $s = 10^3$, the running time of UT_n is always at least one order of magnitude higher than all other approaches, and could not complete in one day for $s > 10^3$.

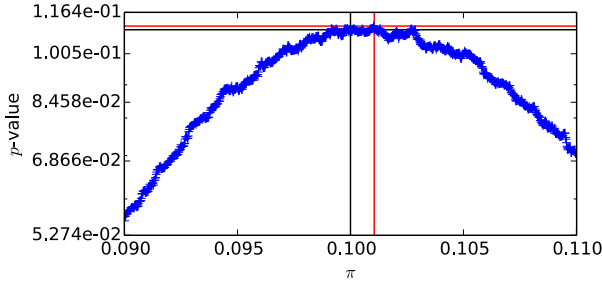


Figure 2: Values of $p_S^{(B)}(\pi)$ for 10^3 equally spaced values of $\pi \in [f(S)(1 - \xi), f(S)(1 + \xi)]$ with $n = 10^3$, $n_1 = 500$, $\sigma(S) = 100$, $\sigma_1(S) = 40$, $\xi = 0.1$. The red vertical line corresponds to $\pi^* = \arg \max_{\pi} \{p_S^{(B)}(\pi)\}$, the black vertical line to $x = f(S)$, the red horizontal to $p_S^{(B)}(\pi^*)$, the black horizontal line to $p_S^{(B)}(f(S))$.

5.1 Results

Quality of the lower bound.

? **QUESTION FROM MATTEO (2/15):** Why are we using $p^{(B)}$ in the following?

Our first experiment aims at evaluating the quality of the lower bound \check{p}_S from (2)). For fixed values of n , n_1 and S (described in Fig. 2), we compute $p_S^{(B)}(\pi)$ varying π over 10^3 equally spaced values in the interval $[f(S)(1 - \xi), f(S)(1 + \xi)]$, with $\xi = 0.1$. The value $p_S^{(B)}(f(S))$ coincides with our lower bound \check{p}_S . Fig. 2 shows the resulting curve: even if $\check{p}_S > p_S$ (therefore $\pi = f(S)$ does not maximize $\phi(\sigma(S), \sigma_1(S), \pi)$), using \check{p}_S provides a principled choice to obtain a very tight lower bound to p_S . Similar results, not shown for space constraints, hold for different choices of n , n_1 , and S .

Running time of SPuMANTE. In Fig. 3.(a), we compare the running times of the SPuMANTE variants UT and UT^* w.r.t. the state-of-the-art by LAMP with Fisher’s exact test, denoted with LAMPF.

? **QUESTION FROM MATTEO (2/16):** In the above sentence, why do we need to say “LAMP with Fisher’s exact test”? What’s the alternative? LAMP with something else?

Contrary to the common belief that unconditional tests are computationally expensive, UT is, in almost all cases, faster than LAMPF. These results stress the efficiency of the upper bound from in Thm. 4.2. The only cases where UT is slower is for small sample sizes ($s \leq 10^4$), and for running times ≤ 10 seconds. In these cases, the time to compute ε in SPuMANTE dominates on the total execution time. For larger sample sizes, UT is faster than LAMPF by up to 1 order of magnitude. UT^* , despite computing \check{p}_S , is also generally faster than LAMPF, thanks to our efficient strategy for computing \check{p}_S (Sect. 4.1.2). UT^* requires more time than LAMP only when $s \leq 10^4$ and for times ≤ 10 s, with the only exception of retail dataset. Thus SPuMANTE provides an efficient strategy for significant pattern mining, even more efficient than the state-of-the-art even if it (correctly) employs an unconditional test.

Statistical power of SPuMANTE. We evaluate the effectiveness of the upper bound from Thm. 4.2 in reporting significant patterns. Fig. 3.(b) displays the number of patterns in the output of UT , UT^* , and LAMPF.

? **QUESTION FROM MATTEO (2/16):** In the following sentence, what’s the *exact* set of results?

The set of results reported by UT^* is always a super-set of the exact set of results, since UT^* uses a lower bound to the exact p -value. UT^* does not provide any guarantee on the FWER of its output, but in our experiments it never reported any false discovery. In all cases UT reports a large set of results, comparable in size with the output of UT^* : combining the confidence intervals and the bound of Thm. 4.2 retains most of the statistical power that would be achieved with the expensive computing of the exact value of p_S .

? **QUESTION FROM MATTEO (2/16):** In the previous sentence, how does the “combining the confidence...” sentence follow from the sentence about the comparable size?

With respect to LAMPF, in all cases, it reports more (in some cases, twice as many) patterns than UT^* .

? **QUESTION FROM MATTEO (2/16):** In the previous sentence, what is the subject? What algorithm is “it”?

Since the p -value computed by Fisher’s exact test can provide an *underestimate* of the true p -value of a pattern (due to the conditional assumptions not being met, see Fig. 1), the additional patterns reported by LAMPF may represent spurious associations.

✓ **To-Do from MATTEO (2/16):** As discussed in our skype call, whether the above is true needs to be investigated.

Lastly, we investigate the impact of using the confidence intervals in SPuMANTE.

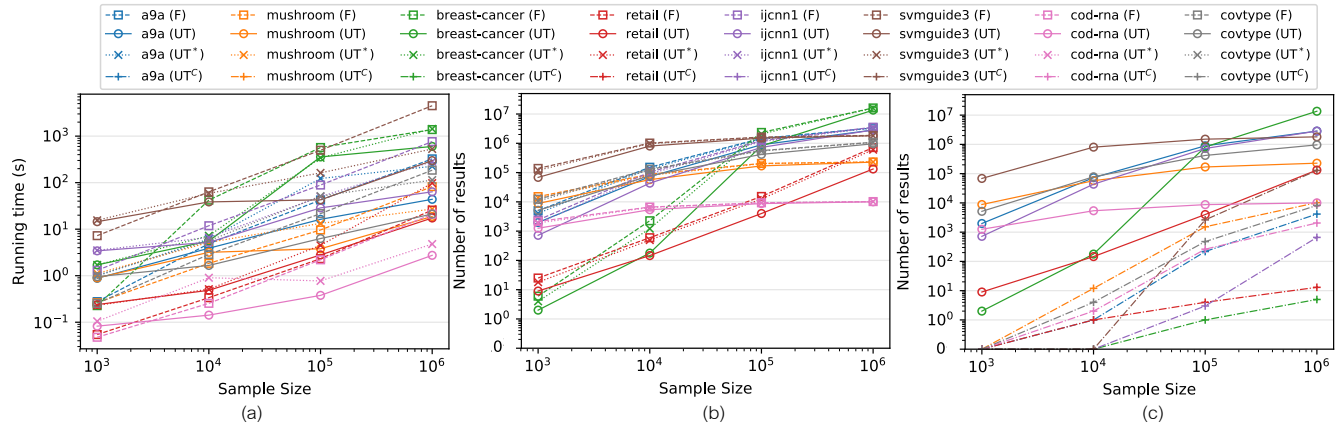


Figure 3: Comparison between LAMPF and SPUMANTE (using UT and UT*) in terms of: (a) running time; (b) number of patterns in output. In (c) we compare the number of patterns in output found with the UT and UT^C variants of SPUMANTE.

NOTE FROM MATTEO (2/16): The above sentence is too vague. As said in the last sentence below, what we want to show is that it is the combination of the good confidence intervals with the good upper bounds that gives us the control of the FWER.

Fig. 3.(c) shows the number of output patterns by UT and by UT^C (the variant of SPUMANTE that only checks whether $C(S) = \emptyset$ to flag a pattern S as significant): the larger the sample, the higher is the number of patterns flagged as significant by UT^C, since ϵ decreases as the sample size grows, so the confidence intervals are more narrow. For the majority of the datasets we considered, a large number of patterns are marked as significant just by checking whether $C(S) = \emptyset$, proving that the use of confidence intervals is a crucial component of SPUMANTE.

6 CONCLUSION

We introduce SPUMANTE, an efficient algorithm to mine significant patterns using our novel unconditional test UT. UT requires fewer assumptions on the data generation process than commonly used conditional tests, such as Fisher's exact test. We prove that SPUMANTE controls the FWER (i.e., the probability of reporting one or more false discoveries) at the desired level α set by the user. Our extensive experimental evaluation shows that SPUMANTE efficiently identifies significant patterns while properly accounting for the stochastic variations in patterns frequencies due to the probabilistic nature of the data generation process.

NOTE FROM MATTEO (2/16): The above sentence feels a bit vague.

There are several future directions of research for our work, including adapting our results to the mining of other significant patterns (e.g., subgraphs) and the use of UT in combination with different procedures for multiple hypothesis correction (e.g., Westfall-Young permutation testing).

REFERENCES

- [1] R. Agrawal, T. Imieliński, and A. Swami. 1993. Mining association rules between sets of items in large databases. *SIGMOD'93*.

- [2] G. A. Barnard. 1945. A new test for 2×2 tables. *Nature* 156 (1945).
- [3] Y. Benjamini and Y. Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Royal Stat. Soc.* (1995).
- [4] R. Berger. 1994. Power comparison of exact unconditional tests for comparing two binomial proportions. *Institute of Statistics Mimeo Series* (1994).
- [5] C. E. Bonferroni. 1936. Teoria statistica delle classi e calcolo delle probabilità. *Pubb. del Regio Istituto Superiore di Scienze Econ. e Comm. di Firenze* 8 (1936).
- [6] R. D. Boschloo. 1970. Raised conditional level of significance for the 2×2 -table when testing the equality of two probabilities. *Statistica Neerlandica* 24 (1970).
- [7] Leena Choi, Jeffrey D. Blume, and William D. Dupont. 2015. Elucidating the foundations of statistical inference with 2×2 tables. *PLoS one* 10, 4 (2015), e0121263.
- [8] Ronald A. Fisher. 1922. On the interpretation of χ^2 from contingency tables, and the calculation of P. *Journal of the Royal Statistical Society* 85, 1 (1922), 87–94.
- [9] W. Hämmäläinen. 2016. New upper bounds for tight and fast approximation of Fisher's exact test in dependency rule mining. *Comp. Stat. & Data Anal.* 93 (2016).
- [10] W. Hämmäläinen and G. I. Webb. 2018. A Tutorial on Statistically Sound Pattern Discovery. *Data Mining and Knowledge Discovery* (2018).
- [11] Zengyou He, Simeng Zhang, and Jun Wu. 2018. Significance-based Discriminative Sequential Pattern Mining. *Expert Systems with Applications* (2018).
- [12] Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics* (1979).
- [13] J. Komiyama, M. Ishihata, H. Arimura, T. Nishibayashi, and S. Minato. 2017. Statistical Emerging Pattern Mining with Multiple Testing Correction. *KDD'17*.
- [14] W. J. Lentz. 1976. Generating Bessel functions in Mie scattering calculations using continued fractions. *Applied Optics* 15 (1976).
- [15] F. Llinares-López, M. Sugiyama, L. Papaxanthos, and K. Borgwardt. 2015. Fast and memory-efficient significant pattern mining via permutation testing. *KDD'15*.
- [16] C. R. Mehta and P. Senchaudhuri. 2003. Conditional versus unconditional exact tests for comparing two binomials. *Cytel Software Corporation* 675 (2003).
- [17] S. Minato, T. Uno, K. Tsuda, A. Terada, and J. Sese. 2014. A fast method of statistical assessment for combinatorial hypotheses based on frequent itemset enumeration. In *ECML-PKDD'14*.
- [18] Laetitia Papaxanthos, F. Llinares-López, D. Bodenham, and K. Borgwardt. 2016. Finding significant combinations of features in the presence of categorical covariates. *NIPS'16*.
- [19] L. Pellegrina and F. Vandin. 2018. Efficient Mining of the Most Significant Patterns with Permutation Testing. *KDD'18*.
- [20] M. Riordan and E. Upfal. 2015. Mining frequent itemsets through progressive sampling with Rademacher averages. *KDD'15*.
- [21] M. Sugiyama, F. Llinares-López, N. Kasenburg, and K. M. Borgwardt. 2015. Significant subgraph mining with multiple testing correction. *ICDM'15*.
- [22] R. E. Tarone. 1990. A modified Bonferroni method for discrete data. *Biometrics* (1990).
- [23] A. Terada, D. duVerle, and K. Tsuda. 2016. Significant Pattern Mining with Confounding Variables. *PAKDD'16*.
- [24] A. Terada, H. Kim, and J. Sese. 2015. High-speed Westfall-Young permutation procedure for genome-wide association studies. *ACM-BCB'15*.
- [25] A. Terada, M. Okada-Hatakeyama, K. Tsuda, and J. Sese. 2013. Statistical significance of combinatorial regulations. *Proc. of the Nat. Acad. of Scien.* 110 (2013).
- [26] F. Vandin, A. Papoutsaki, B. J. Raphael, and E. Upfal. 2015. Accurate computation of survival statistics in genome-wide studies. *PLoS Comp. Bio.* 11 (2015).

- [27] G. I. Webb. 2006. Discovering significant rules. In *KDD'06*.
- [28] G. I. Webb. 2007. Discovering significant patterns. *Machine learning* 68 (2007).
- [29] G. I. Webb. 2008. Layered critical values: a powerful direct-adjustment approach to discovering significant patterns. *Machine Learning* 71 (2008).
- [30] P. H. Westfall and S. S. Young. 1993. Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment. *Wiley Series in Prob. and Stat.* (1993).

A APPENDIX

In this appendix we present the proofs that we could not include in the main text and describe how to reproduce our experimental results.

Missing proofs

The proofs of our results are provided here.

THEOREM A.1 (THM. 4.2 IN THE MAIN TEXT). *It holds*

$$p_S \leq P(\sigma(S), \sigma_1(S) \mid f(S))(n_0 + 1)(n_1 + 1) .$$

PROOF. It holds

$$\begin{aligned} p_S &= \max_{\pi \in C_S} \left\{ \sum_{(x, a) \in T(\sigma(S), \sigma_1(S), \pi)} P(x, a \mid \pi) \right\} \\ &\leq \max_{\pi \in C_S} \{P(\sigma(S), \sigma_1(S) \mid \pi) |T(\sigma(S), \sigma_1(S), \pi)|\} \\ &\leq \max_{\pi \in C_S} \{P(\sigma(S), \sigma_1(S) \mid \pi)\} \max_{\pi \in C_S} \{|T(\sigma(S), \sigma_1(S), \pi)|\} \\ &= P(\sigma(S), \sigma_1(S) \mid f(S))(n_0 + 1)(n_1 + 1). \end{aligned}$$

where in the last step we use Prop. 4.4. \square

PROPOSITION A.2 (PROP. 4.4 IN THE MAIN TEXT). *It holds*

$$\arg \max_{\pi} \{P(x, a \mid \pi)\} = x/n.$$

PROOF. Define the function $f(\pi) = a'(\pi)^b(1 - \pi)^{(c-b)}$ for some constants $a' > 0$, $b = x$, $c = n$. Then

$$\frac{\partial f(\pi)}{\partial \pi} = \frac{a'(1 - \pi)^{(c-b)}\pi^{(b-1)}(c\pi - b)}{\pi - 1} .$$

The only root of $\frac{\partial f(\pi)}{\partial \pi}$ for $\pi \in (0, 1)$ is given by $\pi = \frac{b}{c} = \frac{x}{n} = f(S)$. It is trivial to check that the sign of the second order derivative is always < 0 , and this fact completes the proof. \square

THEOREM A.3 (THM. 4.5 IN THE MAIN TEXT). *Let $C_0(S) \cap C_1(S) = C_S \neq \emptyset$. Then $f(S) \in C_S$.*

PROOF. We prove the result assuming $\sigma_0(S)/n_0 > \sigma_1(S)/n_1$ (the proof for the other case is analogous) and assuming that the confidence intervals have the form provided by AMIRA. Recall that $f(S) = \sigma(S)/n$. $C_0(S) \cap C_1(S) \neq \emptyset$ is equivalent to

$$\frac{\sigma_1(S)}{n_1} + \varepsilon \frac{n}{n_1} \geq \frac{\sigma_0(S)}{n_0} - \varepsilon \frac{n}{n_0} . \quad (6)$$

and that proving $\sigma(S)/n \in C_0(S) \cap C_1(S)$ corresponds to prove that

$$\frac{\sigma_1(S)}{n_1} + \varepsilon \frac{n}{n_1} \geq \frac{\sigma(S)}{n} \quad (7)$$

and

$$\frac{\sigma_0(S)}{n_0} - \varepsilon \frac{n}{n_0} \leq \frac{\sigma(S)}{n} \quad (8)$$

both hold. (6) above is equivalent to

$$\frac{\sigma_0(S)}{n_0} - \frac{\sigma_1(S)}{n_1} \leq \varepsilon n \left(\frac{1}{n_0} + \frac{1}{n_1} \right) . \quad (9)$$

It holds

$$\frac{\sigma(S)}{n} = \frac{\sigma_1(S)}{n_1} + \frac{n_0}{n} \left(\frac{\sigma_0(S)}{n_0} - \frac{\sigma_1(S)}{n_1} \right)$$

and from (9) we derive

$$\begin{aligned} \frac{\sigma(S)}{n} &= \frac{\sigma_1(S)}{n_1} + \frac{n_0}{n} \left(\frac{\sigma_0(S)}{n_0} - \frac{\sigma_1(S)}{n_1} \right) \\ &\leq \frac{\sigma_1(S)}{n_1} + \frac{n_0}{n} \varepsilon n \left(\frac{1}{n_0} + \frac{1}{n_1} \right) \\ &= \frac{\sigma_1(S)}{n_1} + n_0 \varepsilon \left(\frac{1}{n_0} + \frac{1}{n_1} \right) \\ &= \frac{\sigma_1(S)}{n_1} + \varepsilon \left(1 + \frac{n_0}{n_1} \right) \\ &= \frac{\sigma_1(S)}{n_1} + \varepsilon \frac{n}{n_1} \end{aligned}$$

that proves (7).

For (8), it holds

$$\frac{\sigma(S)}{n} = \frac{\sigma_0(S)}{n_0} - \frac{n_1}{n} \left(\frac{\sigma_0(S)}{n_0} - \frac{\sigma_1(S)}{n_1} \right)$$

and from (9) we derive

$$\begin{aligned} \frac{\sigma(S)}{n} &= \frac{\sigma_0(S)}{n_0} - \frac{n_1}{n} \left(\frac{\sigma_0(S)}{n_0} - \frac{\sigma_1(S)}{n_1} \right) \\ &\geq \frac{\sigma_0(S)}{n_0} - \frac{n_1}{n} \varepsilon n \left(\frac{1}{n_0} + \frac{1}{n_1} \right) \\ &= \frac{\sigma_0(S)}{n_0} - n_1 \varepsilon \left(\frac{1}{n_0} + \frac{1}{n_1} \right) \\ &= \frac{\sigma_0(S)}{n_0} - \varepsilon \left(1 + \frac{n_1}{n_0} \right) \\ &= \frac{\sigma_0(S)}{n_0} - \varepsilon \frac{n}{n_0} \end{aligned}$$

that proves (8). \square

PROPOSITION A.4 (PROP. 4.6 IN THE MAIN TEXT). Let $y \in [0, n]$, $b \in [0, \min\{y, n_1\}]$, and $\pi \in (0, 1)$. Define

$$A_1 = \{a_1 : P(a_1 + \lfloor (n_0 + 1)\pi \rfloor, a_1 | \pi) > P(y, b | \pi)\},$$

and define the set

$$A_{0, a_1} = \{a_0 : P(a_1 + a_0, a_1 | \pi) \leq P(y, b | \pi)\}.$$

Then

$$\begin{aligned} &\sum_{(x, a) \in T(y, b, \pi)} P(x, a | \pi) \\ &= \sum_{a_1 \notin A_1} B(a_1, n_1, \pi) + \sum_{a_1 \in A_1} \left(B(a_1, n_1, \pi) \sum_{a_0 \in A_{0, a_1}} B(a_0, n_0, \pi) \right), \end{aligned}$$

where $B(z, h, \pi) = \binom{h}{z} \pi^z (1 - \pi)^{h-z}$ is the probability of obtaining z successes on h independent trials with success probability π .

PROOF. We formulate $\sum_{(x, a) \in T(y, b, \pi)} P(x, a | \pi)$ as

$$\begin{aligned} &\sum_{(x, a) \in T(y, b, \pi)} P(x, a | \pi) \\ &= \sum_{(a_0 + a_1, a_1) \in T(y, b, \pi)} \binom{n_0}{a_0} \binom{n_1}{a_1} (\pi)^{a_0 + a_1} (1 - \pi)^{(n_0 + n_1 - a_0 - a_1)} \\ &= \sum_{(a_0 + a_1, a_1) \in T(y, b, \pi)} B(a_0, n_0, \pi) B(a_1, n_1, \pi) \end{aligned}$$

Let $\lfloor a \rfloor$ denote the closest integer to a . It holds

$$P(a_1 + \lfloor (n_0 + 1)\pi \rfloor, a_1 | \pi) \geq P(a_1 + a_0, a_1 | \pi), \forall a_0 \in [0, n_0].$$

? QUESTION FROM MATTEO (2/16): Why are we redefining the following sets?

Let A_1 be the set of values of a_1 such that there exist at least one contingency table with probability $> P(y, b | \pi)$, that from the above can be defined as

$$A_1 = \{a_1 : P(a_1 + \lfloor (n_0 + 1)\pi \rfloor, a_1 | \pi) > P(y, b | \pi)\}.$$

Let A_{0, a_1} be the set all the values of a_0 , for a given a_1 , such that $P(a_1 + a_0, a_1 | \pi) \leq P(y, b | \pi)$, that is

$$A_{0, a_1} = \{a_0 : P(a_1 + a_0, a_1 | \pi) \leq P(y, b | \pi)\}.$$

It is then easy to show that

$$\begin{aligned} &\sum_{(x, a) \in T(y, b, \pi)} P(x, a | \pi) \\ &= \sum_{(a_0 + a_1, a_1) \in T(y, b, \pi)} B(a_0, n_0, \pi) B(a_1, n_1, \pi) \\ &= \sum_{a_1 \notin A_1} B(a_1, n_1, \pi) + \sum_{a_1 \in A_1} \left(B(a_1, n_1, \pi) \sum_{a_0 \in A_{0, a_1}} B(a_0, n_0, \pi) \right) \end{aligned}$$

\square

Experimental reproducibility

We now describe how to reproduce our experimental results. Code and data are available from <https://goo.gl/sxx8JX>.

Reproducing our simulations. The data and the plot in Fig. 1 can be created with the Python script `fisher_simulations.py` in the `scripts/` folder. The script computes (in parallel) 10^6 random p -values for both Fisher's and Barnard's tests, printing them on a `.csv` file, and then plots them as in Fig. 1. The results of Fig. 2 can be obtained using the `find_max_pi.py` script. All the parameters of the experiments can be modified with appropriate input parameters, or by directly modifying the scripts.

Reproducing our experiments. The code of SPuMANTE and all variants of UT are in the sub-folder `unconditional/`, while the code for LAMPF is in the sub-folder `fisher/`. Inside each folder, the `correct/` directory contains the code for computing the corrected significance threshold δ , while the `enumerate/` directory contains the code to actually compute the significant patterns.

To compile all the software, use the `make` command inside all `correct/` and `enumerate/` sub-folders. Then, also compile AMIRA by running `make` inside the `amira/` folder. A recent version of GCC (e.g., GCC 8.0) is needed to compile AMIRA.

Once everything has been compiled, convenient scripts can be used to run the experiments. In particular, `run_amira.py`, `run_unconditional.py` and `run_fisher.py` automatically execute AMIRA, SPuMANTE, and LAMPF, respectively. These scripts accept a variety of input parameters. In particular, you need to specify a particular dataset and the size of a random sample to create using the flags `-db` and `-sz`. As an example, the command line to

process with SPuMANTE a random sample of 10^3 transactions from the dataset mushroom is

```
run_unconditional.py -db mushroom -sz 1000}
```

and it automatically executes AMIRA and SPuMANTE.

The `run_all_datasets.py` script runs all the instances of SPuMANTE and LAMPF in parallel, and can be used to reproduce all the experiments described in Sect. 5.