Expect the Unexpected – On the Significance of Subgroups

Matthijs van Leeuwen[1] and Antti Ukkonen[2]

[1] Leiden Institute of Advanced Computer Science, Leiden, The Netherlands
[2] Finnish Institute of Occupational Health, Helsinki, Finland
m.van.leeuwen@liacs.leidenuniv.nl, antti.ukkonen@ttl.fi

Abstract. Within the field of exploratory data mining, subgroup discovery is concerned with finding regions in the data that stand out with respect to a particular target. An important question is how to validate the patterns found; how do we distinguish a true finding from a false discovery? A common solution is to apply a statistical significance test that states that a pattern is real iff it is different from a random subset. In this paper we argue and empirically show that this assumption is often too weak, as almost any realistic pattern language specifies a set of subsets that strongly deviates from random subsets. In particular, our analysis shows that one should *expect the unexpected* in subgroup discovery: given a dataset and corresponding description language, it is very likely that high-quality subgroups can —and hence will— be found.

1 Introduction

Subgroup Discovery (SD) [7,19] is concerned with finding regions in the data that stand out with respect to a given target. It has many closely related cousins, such as Significant Pattern Mining [17] and Emerging Pattern Mining [1], which all concern the discovery of patterns correlated with a Boolean target concept. The Subgroup Discovery task is more generic, i.e., it is agnostic of data and pattern types. For example, the target can be either discrete or numeric [5].

A large number of SD algorithms and quality measures have been proposed and it is easy to mine thousands or millions of patterns from data. One question that naturally arises is how to validate these (potential) discoveries: how do we distinguish a 'true' pattern from a 'fluke' that is present in the data by chance and therefore does not represent any true correlation between description and target? In other words, how do we distinguish true from false discoveries?

Statistical testing for pattern mining has received quite some attention over the past decade. Webb was one of the pioneers of this topic and pointed out already in 2007 that the size of the entire pattern space under consideration should be used as Bonferroni correction factor for multiple hypothesis testing [18]. He also observed that when the null hypothesis supposes that the consequent of a rule/pattern is independent of its antedecent, Monte Carlo sampling of the target variable can be used to generate a null distribution. Almost simultaneously, Gionis et al. introduced randomisation testing on Boolean matrices [3].

Permutation tests [4] are a non-parametric approach to hypothesis testing, with the main advantage that p-values are computed from simulations instead of formulas of parametric distributions. This makes them especially suitable for situations where the null-hypothesis cannot be assumed to have a known parametric form. As a result, permutation tests have since become fairly popular in the data mining community and have been used for, e.g., studying classifier performance [15] and statistical testing for subgroup discovery [2].

More recently, several efficient statistical pattern mining methods have been proposed, mostly for rule discovery in Boolean matrices. Hämäläinen, for example, proposed Kingfisher [6], for efficiently searching for both positive and negative dependency rules based on Fisher's exact test. Terada et al. [17] introduced LAMP, which considers a similar setting but employs a smaller Bonferroni correction based on the notion of 'testable patterns'. Recently improvements have been introduced that make LAMP more efficient [13] and versatile [12].

Expect the unexpected The technical assumption underlying most statistical tests for pattern mining is that of *exchangeability* of the observations under the null hypothesis [4]. Simply put, *a priori* the selection of any subset of the data is deemed to be equally likely and the observed patterns are tested against this assumption. In this paper we argue and empirically show that this assumption is often too weak: *in practice almost any pattern language and dataset specify a set of data subsets that strongly deviates from this assumption.*

In particular, we will investigate how likely it is to observe a completely random subset having a large effect size, i.e., a large deviation from the global distribution. As we will see, this probability tends to be very small. Next, we will investigate how likely it is that a given description language, i.e., a set of possible patterns, contains descriptions having large effect size. As we will see, this probability tends to be substantially larger. As a result, we conclude that one should *expect the unexpected* in pattern mining in general and in subgroup discovery in particular: given a dataset and a description language, it is very likely that high-quality subgroups can —and hence will— be found.

One possible conclusion to draw from this is that null-hypothesis significance testing for individual patterns should be used with caution: although it helps to eradicate some very obvious false discoveries, the often-used null hypothesis based on exchangeability may be too weak for the p-values to be useful. Note that we are not the first to warn for the use of significance-based filtering in pattern mining [11], but we are the first to analyse and empirically investigate the effect of the description languages used in pattern mining and their relation to the null hypothesis that assumes random data subsets.

Section 2 introduces the basics of Subgroup Discovery, after which Section 3 presents our approach to quantifying the significance of description languages. More precisely, we will formalise the odds of observing a pattern having a large effect size versus observing a random subset having such a large effect size. As computing these odds exactly is clearly infeasible, we introduce the machinery required for estimating its two components in Sections 4 and 5. We present the empirical analysis in Section 6 and conclude in Section 7.

2 Subgroup Discovery

A dataset $\mathcal{D}$ is a bag of tuples t over the attributes $\{A_1, \ldots, A_m, Y\}$, where the A_i are the *description attributes* A and Y is the target attribute. Each attribute has a Boolean, nominal or numeric domain.

A *subgroup* consists of a *description* and a corresponding *cover*. That is, a *subgroup description* is a pattern S, i.e., a conjunction of conditions on the description attributes. Its corresponding *subgroup cover* is the bag of tuples that satisfy the pattern defined by predicate S, i.e., $C(S) = \{t \in \mathcal{D} \mid t \vDash S\}$. We slightly abuse notation and refer to either the description or its cover using S (depending on context). We use $|S|$ to denote the size of the cover, also called *coverage*. Further, a *description language* L consists of all possible subgroup descriptions, parametrised by maximum depth *maxdepth*, which imposes a maximum on the number of conditions that are allowed in a description.

The Subgroup Discovery task is to find the top-k ranking subgroups according to some quality measure $\varphi : 2^Y \mapsto \mathbb{R}$, which assigns a score to any individual subgroup based on its target values. We consider the well-known Weighted Relative Accuracy (WRAcc), defined as $\varphi_{WRAcc}(S) = \sqrt{|S|}(\mu(S) - \mu)$, where μ is the mean of the target variable (restricted to the tuples in the subgroup cover in case of $\mu(S)$). WRAcc is well-defined for Boolean target attributes by interpreting the proportion of ones as the mean of Boolean values.

To discover high-quality subgroups, top-down search through the pattern space is commonly used. Several parameters influence the search, e.g., a minimum coverage threshold requires subgroup covers to consist of at least *mincov* tuples and the *maxdepth* parameter allows to restrict the size of the search space. During search, the overall top-k subgroups are usually kept as final result.

3 Estimating the Significance of a Description Language

In this section we motivate and formalise the approach that we will take to empirically investigate the significance of subgroups or, more accurately, their description languages. For this, we start by making three important observations.

The first observation, which we already mentioned in the Introduction, is that *the null hypotheses of most statistical tests for pattern mining are based on random subsets of the data*. This is also the case for, for example, the target permutation test proposed by Duivesteijn and Knobbe [2]. The rationale for permuting the target attribute is that this will result in datasets with no meaningful functional relationship between the description attributes and the target. Subgroups found in the actual data should have a higher quality than those found on the permuted data, or otherwise the subgroup is there "by chance".

We will empirically show, however, that given some 'non-random' dataset, almost any description language corresponds to a set of subsets that is very different from the set of random subsets, trivially resulting in very low p-values. Hence, this type of null hypothesis is often too weak and any approach based on this assumption will render many patterns significant; see, e.g., [2,18].

The second observation is that *pattern mining techniques search for the best patterns present in a given description language*. This observation has two immediate consequences. First, one cannot only consider the best found pattern and treat this as 'the result', i.e., as if this were the only observation. Instead, as Webb observed [18], one has to take into account that all patterns in the description language are (implicitly or explicitly) considered and hence apply multiple hypothesis correction, for example Bonferroni correction. Second, this also means that there is essentially no difference between testing 1) a description language in its entirety and 2) the best patterns that were found using search. We therefore deviate from the common approach and investigate the 'significance' of entire description languages rather than that of individual patterns.

Finally, the third observation is that *it is key to distinguish —using appropriate statistical terms— sample size on one hand and effect size on the other hand*. In case of Subgroup Discovery, this implies distinguishing subgroup coverage from the difference in target distribution between the subgroup and the dataset. These two quantities are usually combined by the quality measure for the purpose of ranking the patterns, but —unlike Duivesteijn and Knobbe [2]— we will treat these strictly separately for our analysis. There are two reasons for this: 1) quality measures such as Weighted Relative Accuracy are somewhat arbitrary combinations of coverage and relative accuracy, and using a global quality threshold may therefore result in somewhat arbitrary decisions, 2) larger effect sizes are more likely for smaller sample sizes.

Thus, we need to define sample size and effect size. Sample size is simply subgroup coverage as defined in the previous section. For the purpose of this paper we base the effect size on WRAcc and hence define it as $q(S) = (\mu(S) - \mu)/\sigma$, where μ and σ are the mean and standard deviation of the target variable.

3.1 Description Languages and Accessible Subsets

Although patterns are specified by the descriptions that make up a description language, what really matters for our analysis are the subsets of the data that these descriptions imply, i.e., their corresponding covers. We therefore introduce the notion of *accessible subsets*.

Given a description language L and data $\mathcal{D}$, denote by $X_{L,\mathcal{D}}$ the *set of all possible non-empty subgroup covers* in $\mathcal{D}$ that are *accessible* using L. More formally, we define the set of accessible subsets as

$$X_{L,\mathcal{D}} = \{E \subseteq \mathcal{D} \mid \exists S \in L \text{ s.t. } C(S) = E, |E| > 0\}. \quad (1)$$

While the precise size of $X_{L,\mathcal{D}}$, i.e., $|X_{L,\mathcal{D}}|$, is data dependent, in general it will be (much) smaller than 2^N, with $N = |\mathcal{D}|$. This leads to another observation: *in most practical cases only a fraction of subsets of $\mathcal{D}$ are accessible with L.* For instance, in this paper we define L as all conjunctions of conditions on the description attributes up to a given number of conditions (*maxdepth*). This language can only represent subgroups with a cover that is an intersection of half-spaces in the feature space of $\mathcal{D}$, the number of which is substantially lower than 2^N for most practical datasets.

In general, the size of $X_{L,\mathcal{D}}$ can be regarded as a measure of the *expressiveness* of the language L in the database $\mathcal{D}$; it is more informative than the number of descriptions, as multiple descriptions may imply the same cover. Even more important for our purposes, however, is that it fully specifies the set of subsets that one needs to consider in order to determine the significance of a given combination of description language and dataset.

3.2 On the Significance of Description Languages

Although the task of Subgroup Discovery is to find descriptions of 'large' covers having a 'large' effect size, in general all subsets of the data can be associated with an effect size. Thus, we can introduce notions that quantify how probable it is to observe a certain effect size in either 1) an accessible or 2) a random subset and compare these two probabilities. As noted before, this strongly depends on the sample size, i.e., the number of tuples in the subset. We therefore propose to *compare the collection of accessible subsets with all (random) subsets having the same size distribution to determine if the description language and data together specify subgroups having relatively high effect size.*

Let us first consider the accessible subsets. Let $X_{k,L,\mathcal{D}}$ denote the set of all accessible subsets with coverage at least k, i.e., $|S| \geq k$. Then, we are interested in the probability that one such accessible subset has an effect size larger than θ, i.e., how likely is that event? We will empirically estimate this probability as

$$\Pr(q(S) \geq \theta \mid X_{k,L,\mathcal{D}}) = \frac{|\{S \in X_{k,L,\mathcal{D}} \mid q(S) \geq \theta\}|}{|X_{k,L,\mathcal{D}}|}.$$

Next we are interested in the same probability, but computed for all possible subsets while taking the coverage distribution of the accessible subsets into account. We denote this probability by $\Pr(q(S) \geq \theta \mid \mathcal{D}_k)$. Based on our observations we expect the former probability to be much larger than the latter, as this would indicate that the description language is 'significant' with regard to the target attribute: it contains subgroups that have larger effect sizes than are likely to be observed in random subsets.

We formalise the ratio between these two probabilities as the *odds*, a function of coverage threshold k, minimum effect size θ, description language L and dataset $\mathcal{D}$:

$$\text{odds}(k, \theta, L, \mathcal{D}) = \frac{\Pr(q(S) \geq \theta \mid X_{k,L,\mathcal{D}})}{\Pr(q(S) \geq \theta \mid \mathcal{D}_k)}. \tag{2}$$

Exactly computing the odds is clearly infeasible for any realistic dataset. In the following two sections we will therefore introduce techniques to estimate $\Pr(q(S) \geq \theta)$ for accessible subsets and for all subsets, respectively.

4 Estimating $\Pr(q(S) \geq \theta)$ for Accessible Subsets

We will now introduce a method for estimating the size of $X_{L,\mathcal{D}}$ for $|S| \geq k$, with or without constraint on the effect size θ, so that we can estimate $\Pr(q(S) \geq \theta)$

for accessible subsets. For this we build upon the SPECTRA algorithm [10]; we next provide a brief description but refer to [10] for the details, as we will focus on the modifications needed for our current purposes.

The SPECTRA algorithm SPECTRA is a modification of Knuth's seminal algorithm [8] for estimating the size of a combinatorial search tree, which is based on the idea of sampling random paths from the root of the tree to a leaf. Let $\mathbf{d} = (d_0, \ldots, d_{h-1})$ denote the sequence of branching factors observed along a path from the root (on level 0) to a leaf (on level h). The estimate produced by $\mathbf{d}$, which we call the *path estimate*, is defined as $\hat{e}(\mathbf{d}) = \sum_{i=1}^{h} \prod_{i=0}^{i-1} d_i$.

Since search trees are rarely regular in practice, Knuth's algorithm samples a number of different paths, and uses the average of the $\hat{e}(\mathbf{d})$ values as an estimate of the size of the tree. To use this method for estimating the number of patterns having coverage $\geq k$, we sample a number of paths from the root of the pattern lattice up to (and including) the frequent/infrequent border, compute the path estimates, and take the average of these. At every step the branching factor is given by the number of extensions to the current pattern that are still frequent. We need to make a small modification though, because the patterns do not form a tree but a *lattice*, i.e., each node can be reached via multiple paths. We therefore end up with the following estimator:

$$e(\mathbf{d}) = \sum_{i=1}^{h} \frac{1}{i!} \prod_{i=0}^{i-1} d_i, \tag{3}$$

which includes the normalisation term $1/i!$ to account for the $i!$ possible paths that can reach every node on the i^{th} level.

The SPECTRA$^+$ algorithm We describe two extensions to SPECTRA that allow to estimate both $|X_{k,L,\mathcal{D}}|$ and $|\{S \in X_{k,L,\mathcal{D}} \mid q(S) \geq \theta\}|$.

First, observe that, in the context of frequent itemset mining, the number of accessible subsets is equivalent to the number of closed itemsets [16]. We exploit this observation to estimate $|X_{k,L,\mathcal{D}}|$: we estimate the number of closed subgroups, where a subgroup is closed iff there exists no extension that does not change its cover. Let c_i denote an indicator variable for the pattern at level i that is 1 iff it is closed and 0 otherwise. Then, a variant of Knuth's original method can be used to estimate the desired number[3]:

$$e(\mathbf{d}) = \sum_{i=1}^{h} c_{i+1} \frac{1}{i!} \prod_{i=0}^{i-1} d_i. \tag{4}$$

Second, observe that $q(S) \geq \theta$ can be considered as another constraint that can be evaluated for each individual subgroup. Hence, to estimate $|\{S \in X_{k,L,\mathcal{D}} \mid q(S) \geq \theta\}|$ we use $c_i = \mathbb{I}\{S_i \text{ is closed} \wedge q(S_i) \geq \theta\}$ in combination with Eq.4. Note that this simple yet effective constraint-based estimator can be used for many other types of constraints as well.

[3] Note that we here use a simpler yet more versatile approach for estimating the number of closed patterns compared to the one proposed in [10].

Although these modifications maintain the desirable property that the expected estimate is correct, they do lead to an increase in the variance. This is due to the larger differences between the possible paths, which would require many more samples to mitigate. Another potential solution to this problem, one that is computationally less expensive, is to *bias* the path samples: instead of choosing an extension uniformly at random, one can design an alternative sampling distribution for each pattern extension. We consider two such biased samplers:

Frequency-biased sampler From all frequent extensions, one is randomly chosen proportional to coverage. To adjust the estimates accordingly, each branching factor is now computed as the inverse probability of choosing this extension, i.e., $d_i = \frac{\sum_{S \in freq} |C(S)|}{|C(S_{i+1})|}$, where $freq$ is the set of all frequent extensions considered at level i and S_{i+1} is the chosen one.

Quality-biased sampler The previous sampler has a preference towards sampling larger covers, but this not necessarily imply large effect size. Therefore, in an effort to more accurately estimate $|\{S \in X_{k,L,\mathcal{D}} \mid q(S) \geq \theta\}|$ for large θ, we propose to randomly choose proportional to subgroup quality, i.e., WRAcc. The branching factors are modified accordingly: $d_i = \frac{\sum_{S \in freq} \varphi(S)}{\varphi(S_{i+1})}$.

5 Estimating $\Pr(q(S) \geq \theta)$ for All Subsets

Next we turn our attention to estimating $\Pr(q(S) \geq \theta)$ for arbitrary data subsets. To compute this estimate under the constraint where the random subsets follow the same size distribution as the accessible subsets, as required above, we first use SPECTRA$^+$ to estimate the coverage distribution of accessible subsets using a number of different values of k. Then, we use the estimators discussed in this section separately for every value of k, and re-weight the estimates by the probabilities of the respective k. The final estimate is simply the sum of these.

It is thus enough to define the estimators for a fixed coverage k. First, note that the probability we are interested in is simply the expected value of the indicator function $\mathbb{I}\{q(S) \geq \theta\}$:

$$\Pr(q(S) \geq \theta) = \mathrm{E}_{\mathrm{unif}}[\mathbb{I}\{q(S) \geq \theta\}], \tag{5}$$

where $\mathrm{E}_{\mathrm{unif}}$ denotes that the expected value is computed over the uniform distribution of all subsets with coverage k. Next we discuss three different approaches resulting in four different estimators for $\Pr(q(S) \geq \theta)$.

***Asymptotic* – Normal approximation** Since we have defined $q(S) = (\mu(S) - \mu)/\sigma$, where μ and σ are the mean and standard deviation of the target variable, respectively, a simple but efficient heuristic for $\Pr(q(S) \geq \theta)$ is given by the tail of the standard normal distribution. According to the central limit theorem, we have $\sqrt{k}q(S) \sim \mathcal{N}(0,1)$, when $|S| = k$. Let $\mathrm{cdf}(x)$ denote the cumulative density of the standard normal distribution at x. Now we can use $1 - \mathrm{cdf}(\sqrt{k}\theta)$ as an estimate of $\Pr(q(S) \geq \theta)$. In the experiments we will call this the *asymptotic* estimator.

***Naive* – Sample mean** We can also use the sample mean to estimate the expected value in Eq.5. Given $S_1, \ldots, S_R$, a uniformly drawn sample of R subsets of size k, we have

$$\Pr(q(S) \geq \theta) \approx R^{-1} \sum_{i=1}^{R} \mathbb{I}\{q(S_i) \geq \theta\}. \tag{6}$$

This is guaranteed to be an unbiased and consistent estimate of $\Pr(q(S) \geq \theta)$, but it may require a prohibitively large R when estimating very small probabilities, because in such cases we are unlikely to find random subsets S with effect size above θ in a sample of realistic size. In particular, this will happen for large values of θ. We call this estimator *naive* when reporting experimental results.

***Weighted* – Importance sampling** To obtain good estimates also for large minimum effect size θ, we resort to *importance sampling*. The idea of importance sampling (see e.g. [14]) is to increase the probability of observing the event of interest, in our case $\mathbb{I}\{q(S) \geq \theta\}$, and re-weight these when computing the sample mean so that the resulting expected value is equal to the desired probability. It is easy to show (see Appendix A.1)that the expected value of the indicator function under the uniform distribution (Eq. 5) is equal to the expected value of the indicator function weighted by the coefficient $W(S)$ under a biased sampling distribution, that is,

$$\mathrm{E}_{\mathrm{unif}}[\mathbb{I}\{q(S) \geq \theta\}] = \mathrm{E}_{\mathrm{biased}}[W(S)\mathbb{I}\{q(S) \geq \theta\}]. \tag{7}$$

Here $W(S) = \frac{p(S)}{p'(S)}$, where $p(S)$ and $p'(S)$ are the probabilities to draw S under the uniform and biased distributions, respectively. Now we can estimate $\mathrm{E}_{\mathrm{biased}}[W(S)\mathbb{I}\{q(S) \geq \theta\}]$ by drawing $S_1, \ldots, S_R$ according to p' and computing

$$R^{-1} \sum_{i=1}^{R} W(S_i)\mathbb{I}\{q(S_i) \geq \theta\}. \tag{8}$$

Unlike the basic sample mean estimator of Eq. 6, the weighted variant in Eq. 8 can give good estimates with a much smaller R, i.e., we need fewer samples, provided we have chosen the biased sampling distribution p' appropriately. We must thus define both p', as well as compute the weighting factor $W(S)$.

The biased sampling distribution We first discuss p'. Since we want to mainly draw random subsets that have a high $q(S)$, in an ideal situation $p'(S) \propto q(S)$. A very simple way to achieve an effect similar to this is to draw S using condition-specific probabilities that are proportional to the target values of the possible extensions, i.e., all conditions that are considered to be added to the current description. That is, we use a *weighted sampling without replacement* scheme, where the condition weights are proportional to their target values. More formally, let $p'(u)$ denote the probability to draw the condition u (before other conditions have been drawn), and define

$$p'(u) = \frac{t(u)}{\sum_v t(v)}, \tag{9}$$

where $t(\cdot)$ is the target value of an item. After drawing an item u we set the probability $p'(u)$ to zero, and renormalise the remaining probabilities to sum up to 1. This is repeated until we have drawn k items. This will have the effect that items having a high target value are more likely to be selected into S, and hence $q(S)$ will be biased towards higher values.

Next we discuss the weighting factor $W(S)$. The problem is that to compute $W(S)$ exactly we must compute $p'(S)$ (see also Appendix A.2), but no closed form expression for $p'(S)$ exists under weighted sampling without replacement, and computing $p'(S)$ exactly is infeasible. Hence we consider two heuristics.

Approximating $W(S)$ by assuming a sampling with replacement scheme As a first approximation, we can consider a scheme that corresponds to sampling *with replacement*. In practice we still draw samples without replacement as usual, but compute $W(S) = p(S)/p'(S)$ as if we had used sampling with replacement. This has the upside that all required probabilities have simple and easy to compute closed form expressions. In particular, we have $\bar{p}(S) = \frac{1}{n^k}$ for uniform sampling with replacement, and $\bar{p}'(S) = \prod_{u \in S} p'(u)$ for weighted sampling with replacement. Given these we can approximate $W(S)$ as

$$\bar{W}_1(S) = \left(n^k \prod_{u \in S} p'(u) \right)^{-1} \tag{10}$$

where $p'(u)$ is defined as in Eq. 9. Our first importance sampling estimator, called W_1 below, is thus defined by replacing $W(S)$ in Eq. 8 with $\bar{W}_1(S)$ of Eq. 10 above. This estimator is not guaranteed to converge to the correct expected value, but the experiments show that it still produces reasonable results.

Approximating $W(S)$ by sampling permutations of S Our second approach to approximate $W(S)$ relies on a different technique. As mentioned above, the basic problem of computing $W(S)$ exactly under weighted sampling without replacement is that it would require us to consider the probabilities of all possible permutations in which S could have been drawn. Doing this is clearly infeasible in practice, as there are $k!$ permutations for sets of size k. However, we can compute an approximation of $W(S)$ by using only a small sample of permutations (see also Appendix A.3). Concretely, given a sample of Q permutations of the set S, denoted $\pi^1, \ldots, \pi^Q$, we can estimate $W(S)$ as

$$\bar{W}_2(S) = \left(\frac{1}{Q} \sum_{i=1}^{Q} \Pr(\pi^i) \right)^{-1} \frac{(n-k)!}{n!}, \tag{11}$$

where $\Pr(\pi^i)$ is the probability to draw the permutation π^i under the weighted sampling without replacement scheme. Our second importance sampling estimator, called W_2, can now be defined by replacing $W(S)$ in Eq. 8 with $\bar{W}_2(S)$ of Eq. 11.

Table 1. Datasets according to target type. For each dataset the number of tuples and the number of discrete resp. numeric description attributes are given.

Dataset	*Properties*			*Dataset*	*Properties*		
	$\|\mathcal{D}\|$	$\|disc\|$	$\|num\|$		$\|\mathcal{D}\|$	$\|disc\|$	$\|num\|$
Boolean target				*Numeric target*			
Adult	48842	8	6	Abalone	4177	1	7
Breast cancer	699	0	9	Crime	1994	1	101
Mushroom	8124	22	0	Elections	1846	71	2
Pima	768	0	8	Helsinki	8337	1	22
Spambase	4601	57	0	Housing	506	1	12
Tic-tac-toe	958	9	0	RedWine	1599	0	11
				Wages	534	7	3

6 Experiments

We here present three experiments, of which the first two concern an assessment of the sampling accuracy of the estimators introduced in the previous two sections. After that, we present our empirical analysis of the odds of observing high effect size subgroups on a selection of datasets.

Table 1 presents the datasets that we use for Experiments 2 and 3, which all have either a Boolean or a numeric target. Except for Crime and Elections, which were described in [9], all were taken from the UCI Machine Learning repository[4]. On-the-fly discretisation of numeric description attributes was applied, meaning that 6 equal-size intervals were created upon pattern extension.

6.1 Experiment 1: Estimating $\Pr(q(S) \geq \theta)$ in All Subsets

We discuss an experiment to compare the four estimators of $\Pr(q(S) \geq \theta)$ in all subsets. One difficulty with this is to generate a target variable so that an exact ground truth probability can be calculated. It should also be possible to vary the skew of the resulting distribution, as the estimators may be affected by this.

In this experiment we generate the target variable as follows. Let $\mathbf{x}$ denote a vector of length n. We set the first m elements of $\mathbf{x}$ equal to $v \geq 1$, and let the remaining $n - m$ elements of $\mathbf{x}$ be equal to 1. That is, let $\mathbf{x}[1 : m] = v$, and $\mathbf{x}[(m + 1) : n] = 1$. By varying m and v we can adjust the skew, with lower values of m and higher values of v implying a higher skew. Let μ and σ denote the mean and standard deviation of $\mathbf{x}$, respectively. The ground truth probability for a given θ is now given by

$$\Pr(q(S) \geq \theta_l) = \binom{n}{k}^{-1} \sum_{i=k-l}^{k} \binom{m}{i} \binom{n-m}{k-i}, \tag{12}$$

where $l = \lfloor \frac{k(v-\theta\sigma-\mu)}{v-1} \rfloor$, and $\theta_l = (((k-l)v+l)/k-\mu)/\sigma$ (Appendix A.4). Notice that since we take the floor when computing l, the resulting probability does not

[4] `http://archive.ics.uci.edu/ml/`

Table 2. Parameter values for synthetic target generation used in Experiment 1. Notice that values for skew parameter m and subgroup coverage k are given as fractions of n (the size of the generated target variable). For example, one combination that these give for n, m and k is $n = 2000$, $m = 0.3 \times 2000 = 600$, $k = 0.05 \times 2000 = 100$.

Parameter	Values	Parameter	Values
n	$1000, 2000, 3000, 4000$	k	$0.05 \times n, 0.10 \times n$
m	$0.1 \times n, 0.3 \times n, 0.6 \times n$	θ	$0.2, 0.4, 0.6, 0.8, 1.0$
v	$3, 6, 9$	R	500

Table 3. Mean absolute error (MAE), minimum and maximum error, and recall of the estimators for low θ (left side) as well as high values of θ (right side) when $R = 500$.

	$\theta \geq 0$			$\theta \geq 0.6$		
Estimator	MAE	(min, max) error	Recall	MAE	(min, max) error	Recall
Asymptotic	2.55	(0.004 19.2)	1.00	3.96	(0.035, 19.22)	1.000
Naive	0.18	(0.00, 1.5)	0.26	0.81	(0.677, 0.95)	0.014
W_1	2.76	(0.00, 25.8)	0.84	1.61	(0.00, 11.19)	0.743
W_2	2.67	(0.00, 30.3)	0.84	0.75	(0.00, 15.73)	0.743

correspond exactly to the given θ, but matches a value θ_l that is slightly larger than θ.

We run all our estimators with all possible parameter combinations shown in Table 2. For the naive estimator we use $10 \times R$ samples, while W_1 and W_2 use both exactly R samples. When estimating with W_2 we set $Q = 200$. To get a fair comparison between the importance sampling estimators, we use the same set of R samples for both W_1 and W_2. Notice that computing the exact probability may fail due to problems with numerical computation. Such cases are omitted from consideration. This experiment was run in R 3.3[5] on Linux.

As estimating small probabilities accurately is difficult, we are mainly interested in calculating the correct order of magnitude, i.e., the estimates we report are always base-10 logarithms of $\Pr(q(S) \geq \theta)$. We also report the mean absolute error (MAE) of these over some subsets of the parameter combinations. Moreover, for some inputs one or more of the estimators fail to produce a positive, non-zero estimate. To get an overview of how prevalent this is, we compute the *recall* over all 360 parameter combinations (the two values of R are handled separately) by considering the fraction of cases where an estimate was obtained of those where it was possible to compute the exact probability.

An overview of the results is given in Table 3 for $R = 500$. On the left we show both MAE (as well as the minimum and maximum error) and recall when considering all parameter combinations, and on the right we report the same numbers for cases with $\theta \geq 0.6$. Larger values of θ are often more interesting in practice, as we will show below when studying $\Pr(q(S) \geq \theta)$ in accessible subsets of real data. We observe that the naive estimator has the lowest MAE in every

[5] `https://www.R-project.org`

case. However, it also has a very low recall, meaning that it only rarely produces a useful estimate, but if it does, the estimate is rather accurate. (This is of course what one would expect.) But especially for $\theta \geq 0.6$ the naive estimator is in practice useless due to having almost zero recall. The asymptotic estimator, on the other hand, always has the highest possible recall ($= 1$), while the importance sampling estimators W_1 and W_2 fail to calculate a positive non-zero estimate in about 15 to 25 percent of the cases. When $\theta \geq 0$, the importance sampling estimators perform roughly at the same level as the asymptotic estimator in terms of MAE. However, when considering performance for $\theta \geq 0.6$, we find that the importance sampling estimators, especially W_2, have substantially lower MAE without losing that much in recall. We conclude that for high values of θ the W_2 estimator gives very good results.

6.2 Experiment 2: Estimating $\Pr(q(S) \geq \theta)$ in Accessible Subsets

As explained in Section 3, we estimate $\Pr(q(S) \geq \theta)$ from two components: 1) the number of accessible subsets having $q(S) \geq \theta$), and 2) the total number of accessible subsets. We estimate both components using SPECTRA$^+$. To evaluate the estimators we compare them against probabilities based on exhaustive counts. Although these can be obtained through simply enumerating all patterns and then counting the number of distinct accessible subsets, this is clearly only feasible for relatively small datasets and description languages. In the following we therefore limit our evaluation to datasets and maximum depth settings for which it was possible to enumerate and count all accessible subsets within a day.

We have introduced three variants of SPECTRA$^+$: the *uniform* variant that uses uniform path sampling, the *frequency* variant that uses the frequencies of the refinements to bias the sampling, and the *quality* variant that uses Weighted Relative Accuracy to bias the sampling. On the left side of Table 4 we present a comparison of these three variants when estimating $\Pr(q(S) \geq \theta)$ in the datasets. Top part of the table shows datasets for which the true probability can be computed, bottom part shows cases where only estimates are available. Note that values of θ vary across datasets, as for some datasets there are fewer high effect size subsets than for others. Datasets having a numeric target attribute are shown in *italics*. The description language corresponding to a maximum depth of three is used, and the estimates are computed from 1000 path samples.

We observe that none of the estimators consistently has the highest accuracy. All tend to catch the correct order of magnitude, with the exception of the quality-biased sampler that is clearly the least accurate. Based on this we choose to use the uniform sampling variant for the remaining experiments, as it is the simplest to implement, and the other approaches do not seem to lead to substantial practical benefits[6] when estimating $\Pr(q(S) \geq \theta)$.

Next we investigate the effect of the number of path samples as well as the maximum search depth on the accuracy of the estimates. For this we compare

[6] We note, however, that for the simpler problem of merely counting accessible subsets using the frequency-biased sampler may give more accurate results. These results are omitted due to length restrictions.

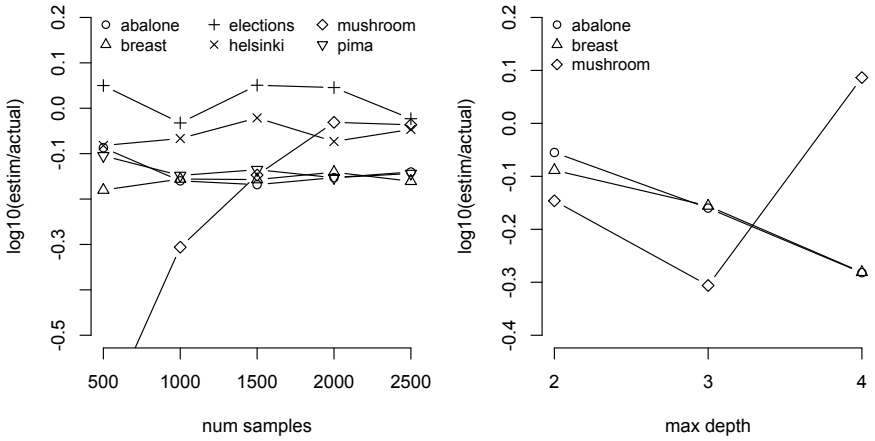

Fig. 1. Estimation accuracy of $\Pr(q(S) \geq 0)$ in accessible subsets using uniform sampling both as a function of the number of path samples (left) and search depth (right) with those datasets and parameter combinations that we were able to compute the exact counts by enumerating all accessible subsets.

estimates obtained using $\{500, 1000, 1500, 2000, 2500\}$ path samples, as well as maximum depths of 2, 3, and 4. The estimation accuracy is shown in Figure 1 for both cases. As we can see from the results, using more path samples does not affect the quality of the estimates in most cases. For most data sets the estimator underestimates the true probability slightly, but the effects are negligible. Given that we are only interested in correctly estimating the order of magnitude, we use 1000 path samples for all remaining experiments. Similar results can be found for maximum depth. Increasing maximum depth may increase the factor by which the path sampling estimator underestimates, but the results are not consistent across all data sets.

6.3 Exp. 3: The Odds of Finding Patterns Having Large Effect Sizes

Lastly, we run the estimators for arbitrary subsets on the real data sets, and compute the odds as defined in Eq. 2. The probability estimates in arbitrary subsets are shown on the right side of Table 4. We have indicated the most reliable estimator in bold. Notice that the importance sampling estimator W_2 tends to produce bad estimates for small values of θ, as we also observed above. In general, if the naive estimator produces a non-zero positive estimate, we take this to be the most reliable value. For small probabilities, however, this will fail, and we must choose between the asymptotic estimator and W_2. We choose W_2 over the asymptotic in situations where the estimators do not deviate substantially.

Finally, we compute the base-10 log odds between 1) the uniform estimator in accessible subsets and 2) the estimator shown in bold for all subsets. We can observe that, for *all datasets*, the probability of observing a high effect size

Table 4. Estimates of $\log_{10}(\Pr(q(S) \geq \theta))$ in accessible subsets (left side) as well as all subsets (right side) obtained with different estimators for the lowest support threshold used ($0.05|\mathcal{D}|$) and a maximum search depth of 3. Target: *Numeric* / Boolean.

	Accessible subsets				θ	**All subsets**			**Odds**
Dataset	Uniform	Frequency	Quality	Exact		Asymp.	Naive	W_2	
Abalone	-0.52	-0.50	-0.19	-0.52	0.4	-10	-Inf	**-9.5**	9.0
Abalone	-1.71	-1.71	-1.56	-1.74	1.0	**-44**	-Inf	-Inf	42.3
Breast cancer	-0.15	-0.13	-0.04	-0.12	0.3	**-1.3**	**-1.3**	Fail	1.2
Breast cancer	-0.37	-0.34	-0.16	-0.29	1.0	**-10**	-Inf	Fail	9.6
Elections	-0.44	-0.60	-0.44	-0.63	0.2	-1.9	**-1.7**	-22.2	1.3
Elections	-1.46	-2.09	-1.51	-1.82	0.5	-6.1	**-4.0**	-19.3	2.5
Helsinki	-0.49	-0.60	-0.16	-0.50	0.4	**-20**	-Inf	**-20**	19.5
Helsinki	-1.08	-1.08	-0.66	-1.16	1.0	**-100**	-Inf	-Inf	98.9
Housing	-0.35	-0.34	-0.15	-0.38	0.4	-1.5	**-1.5**	-1.6	1.2
Housing	-1.23	-1.30	-0.94	-1.24	1.0	-8.1	-Inf	**-6.9**	5.7
Mushroom	-0.74	-0.26	-	-0.27	0.1	**-0.9**	-0.9	-134	0.2
Mushroom	-0.53	-0.76	-	-0.55	0.2	**-5.1**	-Inf	-134	4.6
Pima	-0.44	-0.47	-0.15	-0.50	0.2	-1.4	**-1.6**	-Inf	1.2
Pima	-1.88	-1.51	-1.36	-1.82	0.8	**-7.5**	-Inf	-Inf	5.6
Adult	-0.33	-0.28	-0.19		0.1	**-1.0**	**-1.0**	NA	0.7
Adult	-1.89	-	-2.43		0.2	**-32**	-Inf	NA	30.1
Crime	-0.39	-0.35	-		0.1	-1.0	**-1.0**	-12.4	0.6
Crime	-0.76	-0.73	-0.38		0.4	-5.5	**-4.5**	-11.8	3.7
RedWine	-0.33	-0.34	-		0.1	-0.3	**-0.3**	-0.85	0.0
RedWine	-1.25	-1.22	-0.75		0.6	**-8.3**	-Inf	-Inf	7.1
Spambase	-0.22	-0.28	-0.02		0.2	-3.4	**-3.4**	-97	3.2
Spambase	-0.86	-1.06	-0.59		1.0	**-54**	-Inf	-97	53.1
Tic-tac-toe	-0.28	-0.27	-0.05		0.1	-0.3	**-0.3**	-9.2	0.0
Tic-tac-toe	-1.33	-1.43	-1.06		0.4	-2.8	**-3.1**	-9.2	1.8
Wages	-0.83	-0.78	-0.36		0.4	-2.0	**-1.9**	-1.9	1.1
Wages	-1.99	-2.27	-1.51		1.0	-7.1	-Inf	**-5.5**	3.5

subgroup is *several orders of magnitude* higher in the accessible subsets than what it is in arbitrary subsets having the same size distribution.

7 Conclusions

We have shown that realistic pattern languages contain large numbers of high-quality subgroups; much more so than can be expected from randomly drawn subsets. Although maybe not a surprising result on itself, this does have implications for approaches that aim to eliminate false discoveries. Specifically, statistical tests often rely on the null hypothesis that any subset of the data is equally likely, but this assumption is clearly too weak. That is, one should *expect the unexpected* in pattern mining: given a dataset and a description language, it is very likely that high-quality subgroups can —and hence will— be found.

Note that we do not claim that significance testing is unusable in the context of pattern mining; existing approaches can certainly help to identify obvious false discoveries. Our analysis does demonstrate, however, that it is of interest to investigate significance tests for description languages (rather than individual patterns) that take the inherent structure of accessible subsets into account.

References

1. Dong, G., Zhang, X., Wong, L., Li, J.: CAEP: Classification by aggregating emerging patterns. In: Proceedings of DS'99. pp. 30–42 (1999)
2. Duivesteijn, W., Knobbe, A.: Exploiting false discoveries – statistical validation of patterns and quality measures in subgroup discovery. In: Proceedings of the ICDM'11. pp. 151–160 (2011)
3. Gionis, A., Mannila, H., Mielikäinen, T., Tsaparas, P.: Assessing data mining results via swap randomization. ACM Trans. Knowl. Discov. Data 1(3) (Dec 2007)
4. Good, P.I.: Permutation, parametric and bootstrap tests of hypotheses. Springer Verlag, 3rd edn. (2005)
5. Grosskreutz, H., Rüping, S.: On subgroup discovery in numerical domains. Data Mining and Knowledge Discovery 19(2), 210–226 (2009)
6. Hämäläinen, W.: Kingfisher: an efficient algorithm for searching for both positive and negative dependency rules with statistical significance measures. Knowl. Inf. Syst. 32(2), 383–414 (2012)
7. Klösgen, W.: Advances in Knowledge Discovery and Data Mining, chap. Explora: A Multipattern and Multistrategy Discovery Assistant, pp. 249–271 (1996)
8. Knuth, D.: Estimating the efficiency of backtrack programs. Mathematics of computation 29(129), 122–136 (1975)
9. van Leeuwen, M., Knobbe, A.: Non-redundant subgroup discovery in large and complex data. In: Proceedings of the ECML PKDD'11. pp. 459–474 (2011)
10. van Leeuwen, M., Ukkonen, A.: Fast estimation of the pattern frequency spectrum. In: Proceedings of ECML PKDD 2014. pp. 114–129 (2014)
11. Lemmerich, F., Puppe, F.: A critical view on automatic significance-filtering in pattern mining. In: Proceedings of ECMLPKDD'14 workshop on Statistically Sound Data Mining (2014)
12. Llinares-López, F., Sugiyama, M., Papaxanthos, L., Borgwardt, K.M.: Fast and memory-efficient significant pattern mining via permutation testing. In: Proceedings of KDD 2015. pp. 725–734 (2015)
13. Minato, S., Uno, T., Tsuda, K., Terada, A., Sese, J.: A fast method of statistical assessment for combinatorial hypotheses based on frequent itemset enumeration. In: Proceedings of ECML PKDD 2014. pp. 422–436 (2014)
14. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge Univ.Press (1995)
15. Ojala, M., Garriga, G.C.: Permutation tests for studying classifier performance. Journal of Machine Learning Research 11, 1833–1863 (2010)
16. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering frequent closed itemsets for association rules. In: Proceedings of the ICDT'99. pp. 398–416 (1999)
17. Terada, A., Okada-Hatakeyama, M., Tsuda, K., Sese, J.: Statistical significance of combinatorial regulations. Proceedings of the National Academy of Sciences 110(32), 12996–13001 (2013)
18. Webb, G.I.: Discovering significant patterns. Machine Learning 68(1), 1–33 (2007)
19. Wrobel, S.: An algorithm for multi-relational discovery of subgroups. In: Proceedings of PKDD 1997. pp. 78–87. Springer, Heidelberg (1997)

A Appendix

A.1 Derivation of Equation 7

By definition we have

$$\mathrm{E}_{\mathrm{unif}}[\mathbb{I}\{q(S) \geq \theta\}] = \sum_S p(S)\mathbb{I}\{q(S) \geq \theta\},$$

where the sum is taken over all subsets of size k, and $p(S)$ is the uniform distribution over all such subsets. Let $p'(S)$ denote an alternative probability distribution over subsets. Now we can multiply every term in the sum with $\frac{p'(S)}{p'(S)} = 1$, and obtain

$$\begin{aligned}\sum_S p(S)\mathbb{I}\{q(S) \geq \theta\} &= \sum_S p'(S)\frac{p(S)}{p'(S)}\mathbb{I}\{q(S) \geq \theta\} \\ &= \sum_S p'(S)W(S)\mathbb{I}\{q(S) \geq \theta\} \\ &= \mathrm{E}_{\mathrm{biased}}[W(S)\mathbb{I}\{q(S) \geq \theta\}],\end{aligned}$$

where the weighting factor $W(S) = \frac{p(S)}{p'(S)}$.

A.2 Computing $W(S)$ Exactly

We discuss problems with exact computation of the weighting factor $W(S)$ under sampling without replacement. The main difficulty is in computing $p'(S)$, as we will show. First, let $\pi(S)$ denote the set of all permutations of the items in the set S, and let π denote one of such permutations, and denote by π_i the i:th item of π. Computing the probability of a given $\pi \in \pi(S)$ under weighted sampling without replacement requires updating the item-specific probabilites after each item is drawn as described in the main text. Let $\Pr(\pi_i \mid \pi_1 \ldots \pi_{i-1})$ denote the probability of drawing item π_i given that we have already drawn items $\pi_1, \ldots, \pi_{i-1}$. (This probability is obtained by setting $p'(\pi_1) = \ldots = p'(\pi_{i-1}) = 0$, and normalising the remaining probabilites so that they sum up to 1.) The probability of drawing π is then expressed as

$$\Pr(\pi) = \prod_{i=1}^{k} \Pr(\pi_i \mid \pi_1 \ldots \pi_{i-1}),$$

and thus the exact probability $p'(S)$ under weighted sampling without replacement is given by

$$p'(S) = \sum_{\pi \in \pi(S)} \Pr(\pi) = \sum_{\pi \in \pi(S)} \prod_{i=1}^{k} \Pr(\pi_i \mid \pi_1 \ldots \pi_{i-1}).$$

However, as there are $k!$ permutations in $\pi(S)$, computing the sum above is infeasible for all but very small k.

A.3 Estimating $W(S)$ by Sampling Permutations from $\pi(S)$

We derive the result in Eq. 11 of the main text. We have

$$W(S) = \frac{p(S)}{p'(S)} = \frac{\binom{n}{k}^{-1}}{\sum_{\pi \in \pi(S)} \Pr(\pi)} = \frac{(n-k)!}{n!} \underbrace{\frac{k!}{\sum_{\pi \in \pi(S)} \Pr(\pi)}}_{\mathrm{E}[\Pr(\pi)]^{-1}},$$

where the second term in the rightmost product in fact is equal to the inverse of the expected value $\mathrm{E}[\Pr(\pi)]$ given the uniform distribution over $\pi \in \pi(S)$. (To see this, note that $\mathrm{E}[\Pr(\pi)] = \sum_{\pi \in \pi(S)} \frac{1}{k!} \Pr(\pi) = \frac{1}{k!} \sum_{\pi \in \pi(S)} \Pr(\pi)$.) Thus, we can write

$$W(S) = \mathrm{E}[\Pr(S)]^{-1} \frac{(n-k)!}{n!}.$$

Expressing $W(S)$ in this manner allows us to compute an efficient approximation of it, by using the basic sample mean estimator of $\mathrm{E}[\Pr(S)]$. This will require us to sample a number of permutations, denoted by Q in the main text, but Q will obviously be much smaller than $k!$, making the computation feasible.

A.4 Derivation of Equation 12

Let $\mu(S)$ denote the mean of the target values in the set S, and recall that μ and σ are the mean and standard deviation of the target variable, respectively. By definition, the first m elements of $\mathbf{x}$ are equal to v, while the remaining $n - m$ elements are equal to 1. This means that for an S of size k where $k - l$ values are chosen from the first m elements of $\mathbf{x}$ (and hence are equal to v), and the remaining l values are from the $n-m$ elements of $\mathbf{x}$ that are equal to 1, we must have $\mu(S) = ((k-l)v+l)/k$. For $(\mu(S)-\mu)/\sigma \geq \theta$ to hold for a given θ, at most a certain number of values in S may thus be equal to 1, the others must be equal to v. In other words, l, the number of 1s, must be upper bounded by some value. Finding this upper bound is a matter of substituting $\mu(S) = ((k-l)v+l)/k$ into $(\mu(S) - \mu)/\sigma \geq \theta$ and solving for l This gives $l \leq \frac{k(v-\theta\sigma-\mu)}{v-1}$, which we round to the nearest integer for which the inequality holds.

At least $k - l$ items must thus be chosen from the first m elements of $\mathbf{x}$, while the remaining l elements can be chosen from any of the remaining elements. To count the number of sets that satisfy this, we partition the sets in terms of the number of items they choose from the first m elements. There are $\sum_{i=k-l}^{k} \binom{m}{i}\binom{n-m}{k-i}$ of such sets, and $\binom{n}{k}$ sets of size k in total, giving Equation 12.