UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Head Office: Università degli Studi di Padova
Department: Information Engineering

Ph.D. course in: Information Engineering
Curriculum: Information and Communication Technologies (ICT)
Series: XXXIII, 2021

Rigorous and Efficient Algorithms for Significant and Approximate Pattern Mining

Thesis written with the financial contribution of Fondazione Cariparo

Coordinator: Ch.mo Prof. Andrea Neviani
Supervisor: Ch.mo Prof. Fabio Vandin

Ph.D. student: Leonardo Pellegrina

Rigorous and Efficient Algorithms for Significant and Approximate Pattern Mining

Abstract

Massive amounts of data are generated in all areas of industry and science, such as social networks, biomedical research, finance, and many others. Extracting useful and reliable knowledge from such data is a fundamental task with two challenges: the first is to provide rigorous statistical guarantees on the analysis, controlling the discovery of spurious results. This aspect is particularly important when the validation of the extracted results is expensive (e.g., in biology), or when data-driven decisions need to be performed accurately (e.g., in medicine). Therefore, it is necessary to design methods that are robust to the inherent noise and uncertainty of the data. The second challenge is to design algorithms that scale the computation to the analysis of large datasets, as the ones of interest from the applications. One example is the analysis of data from large genomic experiments, whose availability is rapidly increasing as sequencing costs continue to fall. In such settings, methods capable of computing high quality approximations are often the only available practical option.

Pattern Mining is a key component of Knowledge Discovery that comprise methods that aim at discovering interpretable and useful structure from data. In particular, Significant Pattern Mining aims at extracting patterns with rigorous guarantees in terms of statistical significance of the output, controlling the retrieval of false discoveries. Approximate Pattern Mining, instead, focuses on the fundamental task of computing rigorous, high-quality approximations of collections of interesting Patterns. The objective of this Thesis is to design novel efficient and rigorous techniques for Significant and Approximate Pattern Mining, in three scenarios.

The first scenario we consider is Mining Significant Patterns, such as Itemsets and Subgraphs, from labelled datasets, with guarantees on false discoveries. In this setting we tackle two problems: the problem of efficiently extracting the set of Top-k most Significant Patterns, providing various forms of control of false discoveries, such as bounding the Family-Wise Error Rate ($FWER$), and effectively limiting the output size by focusing on the k most interesting results; and the problem of Mining Significant Patterns with an unconditional statistical test, such as Barnard's exact test, which, as we argue, is more appropriate for Knowledge Discovery applications than traditionally employed conditional tests. For both problems we develop new algorithms and provide theoretical and experimental evidence of their effectiveness.

In the second scenario, we develop a novel method to provide rigorous approximations of the collection of frequent strings of length k, called k-mers, from massive datasets of biological reads, originating from high-throughput sequencing experiments. Counting k-mers is computationally very demanding, and it is one of the first steps in many bioinformatics pipelines. We show with extensive experiments that the

most frequent k-mers can be well estimated by an advanced sampling scheme that analyzes a randomly chosen fraction of the data, resulting in significant computational speedups. Approximating the most frequent k-mers allows to accurately estimate distances between pairs of datasets from metagenomic applications in a fraction of the time required by the exact approaches. We provide a rigorous analysis of our algorithm using the VC dimension, a key concept from Statistical Learning Theory.

In the last scenario, we study the Monte Carlo Empirical Rademacher Average (MCERA), a fundamental data-dependent measure of Statistical Learning Theory of the complexity of families of functions. This important quantity allows computing tight probabilistic uniform deviation bounds between empirical averages of sets of functions from their expectation. First, we develop a general framework to compute the MCERA efficiently, exploiting a combinatorial structure of the set of functions given by a partially-ordered set (*poset*). Our novel approach is general, and it can be applied to a variety of different Pattern Mining problems with this structure, such as Itemsets, Sequences, Subgraphs, and Subgroups. Obtaining sharp uniform convergence bounds has applications in both Significant Pattern Mining and Approximate Pattern Mining; we apply this method to a specific Significant Pattern Mining task, the discovery of True Frequent Patterns, whose goal is to identify Frequent Patterns w.r.t. the generative process of the data. We show that this approach is practical on several real-world dataset, and significantly outperforms recent state-of-the-art methods. Secondly, we study the rate of probabilistic convergence of the MCERA; we prove that it satisfies certain self-bounding properties, important concepts in the theory of concentration inequalities. We show that these properties imply novel variance-dependent convergence bounds, which result in significantly improved bounds on its guaranteed accuracy.

Acknowledgments

Here I am, at the end of another journey. Looking back, this adventure was full of joyful moments, but also many tough challenges. I did my best to learn and become a better man every day, and for this I feel very lucky and thankful to many.

Per prima cosa, vorrei ringraziare Fabio. Mi hai trasmesso il coraggio di intraprendere questa strada, convincendomi che ne sarei stato capace. Ti ringrazio per tutto quello che ho imparato, per i consigli, le giuste critiche, la pazienza, e per come sono cresciuto, sia dal punto di vista scientifico che personale.

Ringrazio Andrea per aver sopportato le mie interminabili presentazioni.

I had the amazing opportunity of visiting Brown; I cannot be grateful enough to Eli for this priviledge. I learned so much from this invaluable experience. Thanks to Matteo, Lorenzo, Cyrus, Benedetto, and Megumi; you all always made me feel welcome, and you truly inspired me with many stimulating discussions. I was lucky to have the best office mate ever; thank you Stefan for all the good times. I will always remember the amazing view from the window of our office. Thanks to Gen, James, and Michele, for the time out together in Providence.

Ringrazio i miei primissimi compagni di laboratorio, Matteo, Emanuele, Michele, che hanno saputo darmi consigli preziosi. E poi Francesco, che in molte occasioni ha saputo condividere esperienza e sane risate. Thanks to the bright fellows Diego, Ilie, Andrea, Davide, Federico, and Thạch, for the spritz, dinners, and many great discussions.

Thanks to all collaborators for their fundamental contributions. I am truly grateful to Jilles and Andrea for their precious comments on this work.

Il ringraziamento più speciale è per Gloria. Se ce l'ho fatta e se sono felice, lo devo a te.

Contents

Chapter 1

Introduction

An overwhelming amount of data is generated at *unprecedented rate* from all areas of science. *Knowledge Discovery* and *Data Mining* are vast and vibrant areas of research, whose aim is to develop methods to *extract useful and reliable information* from such massive data. This fundamental problem poses two major challenges. The first regards the need of providing *rigorous guarantees* on the significance of the results of the analysis; in many applications, reporting spurious insights has extremely expensive consequences. Examples are in biology and medicine, where the results of explorative analyses are validated through follow-up experiments. Therefore, it is critical to design methods that are robust to the intrinsic noise and uncertainty characterising the data. The second is to design methods that *scale such complex computations* to the analysis of the massive datasets generated from applications, such as social networks and genomic experiments. In such settings, performing an exact analysis on the entire data may be too expensive; often the only viable approach is given by the development of algorithms that output provably high quality approximations.

Pattern Mining is an important sub-area of *Knowledge Discovery* and *Data Mining*, whose goal is to discover meaningful and *interpretable structures* from collections of observations. There are many variants of Pattern Mining, depending on the type of data at hand, and the goal of the data analysis task. For example, transactional datasets can be analysed with Itemsets Mining, Subgroup Discovery, Sequential Pattern Mining, while graphlets and subgraphs can be discovered from graphs and collections of graphs. A vast literature of ingenuous algorithms for all such variants have been proposed over the years to tackle many aspects of these fundamental tasks. Pattern Mining methods find widespread applications, ranging from market basket analysis, spam detection, recommendation systems, to the analysis of data generated by biology and medicine.

Significant Pattern Mining comprehends Pattern Mining techniques whose goal is to discover interesting patterns with rigorous guarantees on *false discoveries*; instead, methods for *Approximate Pattern Mining* focus on the efficient computation of high quality approximations of collections of interesting patterns. The goal of this Thesis is to develop new efficient and rigorous methods for the problems of Significant and Approximate Pattern Mining; we will argue that, in some cases, the challenges of both such problems, apparently distinct, are deeply related.

First, we consider the problem of Significant Pattern Mining. This task is an extension of the problem of *Frequent Pattern Mining*, a fundamental primitive of Data Mining whose goal is to identify *highly supported* patterns over the data, for example appearing in a sufficient fraction of all the transactions of a transactional dataset. The main goal of Significant Pattern Mining is, instead, to discover patterns that describe characteristics of the underlying process that generated the data; one example is when each transaction is enriched by a label, and the goal is to identify patterns showing *association* to the labels. Such methods are crucial in many applications, providing additional information w.r.t. mining patterns that are frequent

in the entire dataset: in market basket analysis, it serves to identify itemsets that are purchased more frequently by one group of customers than by another one (e.g., married people vs. singles); in social networks, it finds features characterizing users interested in one specific topic; in biology, it identifies sets of genetic variants appearing more frequently in cancer vs normal tissues, or in one cancer type vs another one. In all applications, it is critical to provide rigorous guarantees on the *statistical significance* of the associations. 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 alone. However, being able to test a single pattern is not enough to provide such guarantees. In fact, when many hypotheses are tested, the risk of reporting spurious associations rapidly increases. As we will discuss, providing guarantees on the overall *false discoveries* without sacrificing the *power* of the method is extremely challenging.

For the task of Significant Pattern Mining, this Thesis work contributes with the following results:

1. In Chapter 3 we consider the problem of mining the *most statistically significant patterns* with rigorous control of the Family-Wise Error Rate ($FWER$). In particular, in analogy with frequent Pattern Mining approaches, we formally define the problem of mining *Top-k Statistically Significant Patterns*. Our definition allows to properly control the size of the output set while providing guarantees on its $FWER$. To solve this problem, we design a novel algorithm, called TopKWY, which provides guarantees on the $FWER$ by using the Westfall-Young permutation testing procedure. TopKWY is based on an exploration strategy similar to the one used by TopKMiner (Pietracaprina and Vandin, 2007), an efficient algorithm to identify the top-k frequent patterns. We prove that the use of such strategy guarantees that, in contrast to previous approaches, TopKWY will never explore *untestable* patterns, candidates with no chance of being significant. Then, we present variants of TopKWY to mine the top-k significant patterns with control of the Generalized Family-Wise Error Rate (g-$FWER$) and the False Discovery Proportion (FDP), more flexible error rates than the $FWER$ that trade an increase in the size of the output with a (potentially) higher, but still controlled, number of false discoveries. These variants lead to an increase in the *statistical power* in situations where the number of results reported controlling the $FWER$ is low. Our extensive experimental evaluation shows that TopKWY enables the extraction of the most significant patterns from large and challenging datasets which could not be analyzed by the state-of-the-art. In addition, TopKWY improves over the state-of-the-art even for the extraction of *all* significant patterns. TopKWY appeared in (Pellegrina and Vandin, 2018, 2020).

2. In Chapter 4 we present SPuManTE, an efficient algorithm for mining signif-

icant patterns from a transactional dataset using an unconditional statistical test. SPuManTE controls the $FWER$, and is based on UT, our novel formulation of an unconditional statistical test for evaluating the significance of patterns; as we will discuss, an unconditional test, such as Barnard's exact test, 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: UT overcomes this issue with a novel algorithm to perform the test efficiently. 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. SPuManTE appeared in (Pellegrina et al., 2019c).

Another challenge of Pattern Mining techniques is how to perform complex analyses of massive datasets. As we discussed, in many situations exact computation is too expensive; Approximate Pattern Mining methods are based on the idea of trading accuracy with large reductions of the computational requirements. A significant challenge in this context is how to quantify the trade-off between such aspects, in particular how to guarantee that the approximated results are sufficiently "close" to the exact ones. A typical example of an Approximate Pattern Mining solution is to perform the analysis on a small, randomly chosen, subsample of a massive dataset. Quantifying the gap between the computed approximation and the exact solution requires to study the effect of the randomness introduced by the sampling process. Techniques from Probability and Statistical Learning Theory are critical to address this challenging aspect.

For this second scenario, we consider the important task of estimating the abundances of all substrings of length k (called k-*mers*) in a set of biological sequences, a fundamental and challenging problem with many applications in computational biology. While several methods have been designed for the exact or approximate solution of this problem, they all require to process the entire dataset, that can be extremely expensive for high-throughput sequencing datasets, given their massive size.

For this problem, the Thesis contributes with the following results:

1. In Chapter 5, we develop, analyze, and test, a sampling-based approach, called `SAKEIMA`, to compute an approximation of the set of frequent k-mers and their frequencies from a high-throughput sequencing dataset, while providing rigorous guarantees on the quality of the approximation. `SAKEIMA` employs an advanced sampling scheme and we show how the characterization of the VC dimension, a core concept from Statistical Learning Theory, of a properly defined set of

functions leads to practical bounds on the sample size required for a rigorous approximation. Our experimental evaluation shows that SAKEIMA allows to rigorously approximate frequent k-mers by processing only a fraction of a dataset. We also show that, while in some applications it is crucial to estimate all k-mers and their abundances, in other situations reporting only *frequent* k-mers, that appear with relatively high frequency in a dataset, may suffice. This is the case, for example, in the computation of k-mers' abundance-based distances among datasets of reads, commonly used in metagenomic analyses. We show that such task can be accelerated using the approximations computed by SAKEIMA. SAKEIMA appeared in (Pellegrina et al., 2019a, 2020b).

Lastly, we address the problem of analyzing *samples* drawn from an *unknown probability distribution*, relevant for Approximate and Significant Pattern Mining problems. We may attribute two meanings to such random samples.

The first meaning is related to Approximate Pattern Mining: the *sample* is intended as a *small random sample of a large dataset*: since it is often impossible or extremely expensive to process massive datasets, it is then reasonable to mine only a small random sample that fits into the main memory of the machine. In many applications, an high-quality approximation of the set of most relevant patterns can be efficiently computed from a sample, with often large reductions of computational requirements. As the sampling process introduces noise, to obtain desirable probabilistic guarantees on the quality of the approximation one must study the *trade-off* between the *size of the sample* and the *quality of the approximation*.

The second meaning is *sample* as a *sample from an unknown data generating distribution*, and it is related to the problem of Significant Pattern Mining: in this case, the whole dataset is seen as a collection of samples from an unknown distribution, and the goal of mining patterns from the available dataset is to discover knowledge about the distribution.

These two meanings of "sample" and distribution are not truly distinct, because also in the first case the goal is to approximate an unknown distribution from a sample, thus falling back into the second case. In both scenarios, advanced probabilistic concepts and results from from Statistical Learning Theory are fundamental to address this problem.

For this scenario, we contribute to novel methods that, as we discuss, find applications in both Approximate Pattern Mining and Significant Pattern Mining:

1. In Chapter 6 we present MCRapper, an algorithm for the efficient computation of *Monte-Carlo Empirical Rademacher Averages (n-MCERA)* for families of functions exhibiting poset (e.g., lattice) structure, such as those that arise in many Pattern Mining tasks. The n-MCERA is a fundamental data-dependent concept of Statistical Learning Theory that quantify the complexity of sets of functions. The use of the n-MCERA allows to compute tight upper bounds to the maximum deviation of sample averages from their expectations, much

sharper compared to methods based on looser approaches (i.e., that bounds the Rademacher Complexity through tools such as Massart's lemma). MCRapper can be used to find both statistically-significant functions (i.e., Significant Patterns) when the available data is seen as a sample from an unknown distribution, and approximations of collections of high-expectation functions (e.g., Frequent Patterns) when the available data is a small sample from a large dataset. The strategy of MCRapper is based on novel upper bounds to the discrepancy of the functions to efficiently explore and prune the search space, a technique borrowed from Pattern Mining itself. To show the practical use of MCRapper, we employ it to develop an algorithm TFP-R for the task of True Frequent Pattern (TFP) mining. TFP-R gives guarantees on the probability of including any false positives (precision) and exhibits higher statistical power (recall) than existing methods offering the same guarantees. We evaluate MCRapper and TFP-R and show that they outperform the state-of-the-art for their respective tasks. MCRapper appeared in (Pellegrina et al., 2020a).

2. In Chapter 7 we derive novel variance-dependent concentration bounds for the n-MCERA, whose convergence rates depend on characteristic quantities of the set of functions under consideration, such as the *empirical wimpy-variance*. Such bounds result in a significantly improved trade-off between the guaranteed accuracy of the estimate of the n-MCERA and the number n of Monte Carlo trials to perform. Our proofs rely on the framework of *self-bounding functions*, important notions of the theory of concentration inequalities; our results follow from the sharp exponential concentration inequalities that self-bounding functions have been shown to satisfy. Such new bounds are relevant and directly applicable to all methods based on the n-MCERA. Then, we also show that the Supremum Deviations (SDs) between empirical averages and their expectations are also self-bounding, for appropriate constants that depend on the maximum and minimum expected values of the functions; consequently, we derive novel concentration inequalities for the SDs, that may be of independent interest. The contributions described in this Chapter appear at (Pellegrina, 2020).

Chapter 2

Background

In this Chapter we introduce the notation and the main concepts we will use for the rest of the Thesis. In Section 2.1 we introduce the preliminary definitions on Pattern Mining, and we present the problems of Frequent and Interesting Pattern Mining, and compare them to the task of Significant Pattern Mining. In Section 2.2 we introduce the framework of Statistical Hypothesis Testing and the problem of Multiple Hypothesis Testing. We introduce methods for Significant Pattern Mining that efficiently leverage fundamental correction procedures for Multiple Hypothesis Testing in Section 2.3. In Section 2.4 we introduce fundamental concepts of Statistical Learning Theory, and in Section 2.5 present recents methods based on their application to Approximate Pattern Mining.

2.1 Pattern Mining

2.1.1 Preliminary Definitions

We consider a set of real-valued *functions* $\mathcal{F}$ from a *domain* $\mathcal{X}$ to an interval $[a, b] \subset \mathbb{R}$. A *set of samples* $\mathcal{S}$ and a *dataset* $\mathcal{D}$ are defined as multisets $\{s_1, \ldots, s_n\}$ of n samples s_i, where each $s_i \in \mathcal{S}$ (or $\in \mathcal{D}$) is a member of $\mathcal{X}$. There are many possible definitions for $\mathcal{X}$, depending on the Knowledge Discovery task to be performed; we now describe the ones we consider in this work.

In many cases, such as in Frequent Pattern Mining, $\mathcal{X}$ is composed by the set of all possible *transactions* $\mathcal{T}$. In *Frequent Itemsets Mining* (Agrawal et al., 1993), given a set of items $\mathcal{I}$, $\mathcal{T}$ is defined as the set of all possible subsets of $\mathcal{I}$. Other definitions of $\mathcal{T}$ we will discuss are sets of sequences or labelled graphs.

In other settings, such as Interesting or Significant Pattern Mining, the transactions $t \in \mathcal{T}$ are also enriched by *target labels* from their respective domain $\mathcal{L}$. In such cases, $\mathcal{X} = \mathcal{T} \times \mathcal{L}$, and therefore the samples $s_i \in \mathcal{S}$ are pairs (t, ℓ) with $t \in \mathcal{T}$ and $\ell \in \mathcal{L}$. We will focus our attention to $\mathcal{L}$ composed by two binary labels $\mathcal{L} = \{\ell^0, \ell^1\}$, even if many of the contributions and methods we present can be extended to more general cases.

In Pattern Mining, it is assumed to have a language $\mathfrak{L}$ containing the patterns of interest. For example, in Itemsets Mining, $\mathfrak{L}$ is the set of all possible *itemsets*, i.e., all non-empty subsets of $\mathcal{I}$, while in Sequential Pattern Mining (Agrawal and Srikant, 1995), $\mathfrak{L}$ is the set of sequences, and in Subgroup Discovery (Klösgen, 1992), $\mathfrak{L}$ is defined by the user as the set of patterns of interest.

In all these cases, for each pattern $\mathcal{P} \in \mathfrak{L}$, we define a function $f_\mathcal{P} \in \mathcal{F}$, such that $f_\mathcal{P}(s)$ denotes the "value" of the pattern $\mathcal{P}$ on the sample $s \in \mathcal{S}$. Therefore, the family of functions $\mathcal{F}$ is the set $\mathcal{F} = \{f_\mathcal{P} : \mathcal{P} \in \mathfrak{L}\}$. As an example, in Frequent Pattern Mining $f_\mathcal{P}$ are indicator functions that map $\mathcal{X}$ to $\{0, 1\}$ so that $f_\mathcal{P}(s) = 1$ iff $\mathcal{P}$ "matches" or "is found in" s, and 0 otherwise; more formally, for an itemset $\mathcal{P}$, $f_\mathcal{P}(s)$ is given by $f_\mathcal{P}(s) = \mathbb{1}[\mathcal{P} \subseteq s]$, where $\mathbb{1}[\cdot]$ is an indicator functions equal to 1 when its argument is true, 0 otherwise. For the case of *high-utility itemset*

mining (Fournier-Viger et al., 2019), the value of $f_{\mathcal{P}}(s)$ would be the utility of $\mathcal{P}$ in the sample s. In the case of Subgroup Discovery and, more generally, in Interesting Pattern Mining, $f_{\mathcal{P}}(s)$ would correspond to a *quality measure* of the pattern $\mathcal{P}$ w.r.t. the labelled transaction $s \in \mathcal{T} \times \mathcal{L}$. Similar reasoning also applies to patterns on graphs, such as *subgraphs* (Jiang et al., 2013) and *graphlets* (Ahmed et al., 2015).

For any definition of $f_{\mathcal{P}}$, an important quantity is its average value $\mathsf{a}_{f_{\mathcal{P}}}(\mathcal{S})$ computed over the elements of $\mathcal{S}$:

$$\mathsf{a}_{f_{\mathcal{P}}}(\mathcal{S}) = \frac{1}{n}\sum_{i=1}^{n} f_{\mathcal{P}}(s_i) \quad .$$

In the next Section we discuss in more details the problems of Frequent and Interesting Pattern Mining, and we discuss how they differ from the task of Significant Pattern Mining.

2.1.2 Frequent and Interesting Pattern Mining

Frequent Pattern Mining

Frequent Pattern Mining is one of the fundamental primitives in Data Mining, with applications in a large number of domains, ranging from market basket analysis to biology and medicine (Han et al., 2007). The goal of Frequent Pattern Mining (Agrawal et al., 1993) is to discover patterns that are observed with high frequency over a set of data. Therefore, such patterns are considered to be interesting as "highly supported" by the observations at hand.

For a given language $\mathfrak{L}$ and a sample $\mathcal{S}$, the measure of interest of this task is the *frequency* of the pattern $\mathcal{P} \in \mathfrak{L}$ in $\mathcal{S}$, that is defined as the fraction of samples $s \in \mathcal{S}$ where $\mathcal{P}$ is found; we will denote the event "$\mathcal{P}$ is found in s" for general patterns by "$\mathcal{P} \subseteq s$". By defining $f_{\mathcal{P}}(s) = \mathbb{1}[\mathcal{P} \subseteq s]$ as an indicator function equal to 1 when this happens, and 0 otherwise, then the average value $\mathsf{a}_{f_{\mathcal{P}}}(\mathcal{S})$ of $f_{\mathcal{P}}$ on $\mathcal{S}$ is exactly the frequency of $\mathcal{P}$ on $\mathcal{S}$. Another important quantity is the *support* $\mathsf{z}_{\mathcal{S}}^{\mathcal{P}}$ of $\mathcal{P}$ in $\mathcal{S}$, that is the number of $s \in \mathcal{S}$ such that $\mathcal{P} \subseteq s$:

$$\mathsf{z}_{\mathcal{S}}^{\mathcal{P}} = \sum_{i=1}^{n} f_{\mathcal{P}}(s_i) = n\mathsf{a}_{f_{\mathcal{P}}}(\mathcal{S}) \quad .$$

Therefore, for a given language $\mathfrak{L}$, a sample $\mathcal{S}$, and a *frequency threshold* $\theta \in [0,1]$, we define the set of Frequent Patterns $\mathsf{FP}(\mathfrak{L}, \mathcal{S}, \theta)$ as

$$\mathsf{FP}(\mathfrak{L}, \mathcal{S}, \theta) = \{(\mathcal{P}, \mathsf{a}_{f_{\mathcal{P}}}(\mathcal{S})) : \mathcal{P} \in \mathfrak{L}, \mathsf{a}_{f_{\mathcal{P}}}(\mathcal{S}) \geq \theta\} \quad .$$

The computation of the set of Frequent Patterns $\mathsf{FP}(\mathfrak{L}, \mathcal{S}, \theta)$ is often extremely challenging because of the massive size of the data at hand and the complex types of languages of interest to the analysis; for these reasons, this has been a central problem

in Data Mining and in Knowledge Discovery, and many ingenious algorithms were developed to address it efficiently (see (Han et al., 2007) for several references).

One important property exploited by such methods is the *anti-monotonicity* of the frequencies of the patterns. Given two patterns $\mathcal{P}_1, \mathcal{P}_2 \in \mathfrak{L}$, we say that $\mathcal{P}_1$ is an *ancestor* of $\mathcal{P}_2$ and, equivalently, that $\mathcal{P}_2$ if a *child* of $\mathcal{P}_1$ if the following holds:

$$\mathcal{P}_2 \subseteq s \implies \mathcal{P}_1 \subseteq s \ , \ \forall s \in \mathcal{X} \ . \tag{2.1}$$

As an example, when $\mathcal{P}_1$ and $\mathcal{P}_2$ are itemsets, this translates in the subset relation $\mathcal{P}_1 \subseteq \mathcal{P}_2$. An immediate consequence of 2.1 is that

$$\begin{aligned} f_{\mathcal{P}_1}(s) \geq f_{\mathcal{P}_2}(s) \ \ , \ \ \forall s \in \mathcal{X} \ , \\ \text{and} \quad \mathsf{a}_{f_{\mathcal{P}_1}}(\mathcal{S}) \geq \mathsf{a}_{f_{\mathcal{P}_2}}(\mathcal{S}) \ \ , \ \ \forall \mathcal{S} \in \mathcal{X}^n \ . \end{aligned}$$

For this reason, if any pattern $\mathcal{P}_1$ is not frequent, i.e. $\mathsf{a}_{f_{\mathcal{P}_1}}(\mathcal{S}) < \theta$, then one can exclude all of its children from consideration, as $\mathsf{a}_{f_{\mathcal{P}_2}}(\mathcal{S}) \geq \theta$ can never hold. This lead to design exploration techniques, such as the algorithm Apriori (Agrawal et al., 1993), of $\mathfrak{L}$ that can prune large portions of the search space when such condition is verified. We may observe that such pairwise relationship between the elements of $\mathfrak{L}$ forms a structure and an ordering over $\mathfrak{L}$; as we will formalize in Chapter 6, many pattern languages can be organized into a partially ordered set, or *poset*, to fully exploit the advantages of this key property also for more general functions.

Since the size of $\mathsf{FP}(\mathfrak{L}, \mathcal{S}, \theta)$ can be extremely large, in particular for small values of θ, and that identifying an appropriate frequency threshold to limit the number of Frequent Patterns is challenging, methods to identify restricted classes of patterns, such as Closed Patterns (Pasquier et al., 1999) or Maximal Patterns (Bayardo Jr, 1998), have been designed. Methods that directly limit the number of patterns by reporting the k most Frequent Closed Patterns have been designed (Han et al., 2002; Pietracaprina and Vandin, 2007) as well. We also note that sophisticated techniques have been proposed to mine diverse and non-redundant sets of interesting patterns (Van Leeuwen and Knobbe, 2012; Vreeken et al., 2011; Knobbe and Ho, 2006; Kalofolias et al., 2017).

Interesting Pattern Mining

Another important task in Data Mining and Knowledge Discovery is Interesting Pattern Mining from a labelled dataset. As discussed in the previous Section, in this scenario the samples $s \in \mathcal{S}$ are enriched by a target label, and thus are pairs (t, ℓ) such that $t \in \mathcal{T}$ and $\ell \in \mathcal{L}$. In this setting, the goal is to discover patterns that are *associated* to the values of the target. One example may be to discover sets of items bought together by customers of a given type, or somatic mutations that are associated to a clinical variable.

First, we denote the multisets of samples $\mathcal{S}^0$ with labels ℓ^0 and $\mathcal{S}^1$ with labels ℓ^1

$$\mathcal{S}^0 = \left\{ t : \left(t, \ell^0\right) \in \mathcal{S} \right\} \ , \ \mathcal{S}^1 = \left\{ t : \left(t, \ell^1\right) \in \mathcal{S} \right\} \ ,$$

and denote their respective sizes $n_0 = |\mathcal{S}^0|$ and $n_1 = |\mathcal{S}^1|$. W.l.o.g., we will assume $n_1 \leq n_0$. Other quantities of interest are the average values $\mathsf{a}_{f_\mathcal{P}}(\mathcal{S}^0)$ of $f_\mathcal{P}$ over elements of $\mathcal{S}^0$ (or $\mathcal{S}^1$), and the support $\mathsf{z}^\mathcal{P}_{\mathcal{S}^0}$ of $\mathcal{P}$ in $\mathcal{S}^0$ (or $\mathcal{S}^1$):

$$\mathsf{a}_{f_\mathcal{P}}\left(\mathcal{S}^0\right) = \frac{1}{n_0} \sum_{t \in \mathcal{S}^0} f_\mathcal{P}(t) \ \ , \ \ \mathsf{z}^\mathcal{P}_{\mathcal{S}^0} = n_0 \mathsf{a}_{f_\mathcal{P}}\left(\mathcal{S}^0\right) \ .$$

If, as before, $f_\mathcal{P}(t) = \mathbb{1}[\mathcal{P} \subseteq t]$, $\mathsf{a}_\mathcal{P}(\mathcal{S}^0)$ is simply the *frequency* of $\mathcal{P}$ in $\mathcal{S}^0$. In this case, a popular quality measure, that we denote by $\mathsf{q}_\mathcal{S}(\mathcal{P})$, quantifies the difference between the proportion of labels $\mathsf{z}^\mathcal{P}_{\mathcal{S}^1}/\mathsf{z}^\mathcal{P}_\mathcal{S}$ over samples $(t, \ell) \in \mathcal{S}$ such that $\mathcal{P} \subseteq t$ and the proportion n_1/n of labels in the entire data $\mathcal{S}$:

$$\mathsf{q}_\mathcal{S}(\mathcal{P}) = \mathsf{a}_\mathcal{P}(\mathcal{S}) \left[\frac{\mathsf{z}^\mathcal{P}_{\mathcal{S}^1}}{\mathsf{z}^\mathcal{P}_\mathcal{S}} - \frac{n_1}{n} \right] \ .$$

Patterns with values of $\mathsf{q}_\mathcal{S}(\mathcal{P})$ far from 0 show an association to one of the target labels of $\mathcal{L}$: samples containing $\mathcal{P}$ have a different proportion of labels than the proportion of labels over the entire data. Thus, one may define the set of Interesting (or High Quality) Patterns $\mathsf{HQP}(\mathfrak{L}, \mathcal{S}, \theta)$ as, for example,

$$\mathsf{HQP}(\mathfrak{L}, \mathcal{S}, \theta) = \{(\mathcal{P}, \mathsf{q}_\mathcal{S}(\mathcal{P})) : \mathcal{P} \in \mathfrak{L}, |\mathsf{q}_\mathcal{S}(\mathcal{P})| \geq \theta\} \ .$$

The quality score $\mathsf{q}_\mathcal{S}(\mathcal{P})$ is widely used[1] in *Subgroups Discovery* (Herrera et al., 2011; Atzmueller, 2015), that is the task of mining high-quality patterns from a user-defined language $\mathfrak{L}$. We remark that $\mathsf{q}_\mathcal{S}(\mathcal{P})$ does not enjoy the same monotonicity property of $\mathsf{a}_{f_\mathcal{P}}(\mathcal{S})$; thus, the task of mining Interesting Patterns is often more involved than the (already very challenging) Frequent Pattern Mining problem.

While formally valid, such quality scores generally do not have a direct statistical interpretation, and therefore it is not always possible to directly assess the statistical properties of $\mathsf{HQP}(\mathfrak{L}, \mathcal{S}, \theta)$; this hinders the often important option to provide rigorous guarantees on the presence of *false positives* in the output, where *false positives* are patterns that are flagged as interesting only as consequence of random fluctuations and not because of a true association with the labels.

Methods for Significant Pattern Mining instead relies on the framework of Statistical Hypothesis Testing to assess the association between the patterns and the target label, in particular to provide rigorous statistical guarantees on the discovery of false positives.

[1] $\mathsf{q}_\mathcal{S}(\mathcal{P})$ is often referred to as 1-quality of $\mathcal{P}$ on $\mathcal{S}$.

Significant Pattern Mining

The goal of *Significant Pattern Mining* (Dong and Bailey, 2012; Hämäläinen and Webb, 2019; Pellegrina et al., 2019b) is to identify patterns having *significant statistical association* with one of the class labels. Significance is commonly assessed using a *statistical test*, which provides a p-value quantifying the probability that the association observed in real data arises due to chance alone.

In the next Section we introduce the framework of Statistical Hypothesis Testing and the issue of Multiple Hypothesis Testing, that arises in Significant Pattern Mining.

2.2 Statistical and Multiple Hypothesis Testing

In Section 2.2.1 we formally define the framework of Statistical Hypothesis Testing; in Section 2.2.2 we describe the problem of Multiple Hypothesis Testing, and in Section 2.2.3 we introduce the most widely used methods to tackle it by bounding the Family-Wise Error Rate ($FWER$), a useful metric to control the discovery of spurious results. We present other error rates, such as the False Discovery Proportion (FDP) and the False Discovery Rate (FDR), in Section 2.2.4.

2.2.1 Statistical Hypothesis Testing

Let H denote a *null hypothesis*, representing the default theory of "nothing interesting" for a question of interest. For example, the null hypothesis $H_{\mathcal{P}}$ of a given pattern $\mathcal{P}$ when transactions have labels would correspond to the hypothesis $H_{\mathcal{P}}$ = "$\mathcal{P}$ is not associated to the labels". A set of observations $\mathcal{S}$ is used to test the validity of H by computing a *test statistic* $\hat{T}_H(\mathcal{S})$, which is a function of H computed on $\mathcal{S}$. Denote by T_H the random variable describing the value of the test statistic $\hat{T}_H(\mathcal{S})$ under the null hypothesis H. From T_H, we can compute a p-value $p_H(\mathcal{S})$, defined as the probability of observing an outcome for T_H that is *equally or more extreme* than $\hat{T}_H(\mathcal{S})$, under the assumption that the null hypothesis H is true:

$$p_H(\mathcal{S}) = \Pr_{\mathcal{S}}\left(\text{“}T_H \text{ more extreme than } \hat{T}_H(\mathcal{S})\text{”} \mid H\right) \quad .$$

The set of outcomes that should be considered "more extreme" depends on the goal of the test, and on the type of data at hand, as we will clarify later in this Section.

A common approach to test H is to compare $p_H(\mathcal{S})$ against a *significance threshold* α: if $p_H(\mathcal{S}) \leq \alpha$, H is *rejected* together with its corresponding default theory of "nothing interesting"; otherwise, H is *not rejected*, as there is no sufficient evidence to reject the default theory. By definition of $p_H(\mathcal{S})$, we have that the probability, taken w.r.t. $\mathcal{S}$, of rejecting H under the assumption that the null hypothesis H is

true is not larger than α:

$$\Pr_{\mathcal{S}}\left(p_H\left(\mathcal{S}\right) \leq \alpha \mid H\right) \leq \alpha \ .$$

Equivalently, under the null hypothesis, $p_H\left(\mathcal{S}\right)$ is a random variable that follows a Uniform Distribution[2] with support in $[0, 1]$; consequently, the probability of making a *Type I* error, i.e. making a *false discovery*, is upper bounded by α.

While our main goal is to bound Type I errors, limiting the risk of rejecting null hypotheses, it is also important to take into account other considerations: for example, a *Type II* error is made when the hypothesis H is accepted as a null hypothesis when it should not, and H results in a *false negative*. The more permissive a procedure is in rejecting hypothesis, the higher is its risk of incurring in Type I errors; on the other end, very a restrictive criteria incurs in many Type II errors. Therefore, there is a clear trade-off between false discoveries and false negatives: it is important to balance Type I and II errors, depending on the particular application.

The definition of the p-value $p_H\left(\mathcal{S}\right)$ depends on the assumptions we make on the distribution of T_H; procedures that describe the computation of an appropriate p-value for a particular set of assumptions are typically denoted by *Statistical Tests*.

As discussed in the previous Section, in Significant Pattern Mining the goal is to test the association between the occurrence of patterns and a target label. We can do so by considering appropriate random variables that model the distribution of the p-value $p_H\left(\mathcal{S}\right)$. For the setting of Significant Pattern Mining, it is sufficient to focus on testing the association of pairs of binary random variables, defined over samples (t, ℓ): the first, given by the indicator functions $\mathbb{1}\left[\mathcal{P} \subseteq t\right]$ and $\mathbb{1}\left[\mathcal{P} \not\subseteq t\right]$, describes the occurrence of the pattern over the samples; the second, given by the indicator functions $\mathbb{1}\left[\ell = \ell^0\right]$ and $\mathbb{1}\left[\ell = \ell^1\right]$, models the labelling of the samples. A useful representation of the distribution of a pattern over the data and over the labels is given by a 2×2 *contingency table*. Table 2.1 shows a contingency table built for describing the appearance of a pattern $\mathcal{P}$ over $\mathcal{S}^0$ and $\mathcal{S}^1$. The 4 innermost entries of the table contain the counts of the samples of $\mathcal{S}$ that satisfy the conditions given by the corresponding column and row headers; for example, the entry $\mathsf{z}_{\mathcal{P}}(\mathcal{S}^1)$ counts the number of samples $(t, s) \in \mathcal{S}^1$ such that $\mathcal{P} \subseteq t$. Row and column totals (i.e., n_1 and $\mathsf{z}(\mathcal{P})$) are also called *marginals* of the contingency table.

In what follow, we introduce the most widely used statistical tests to assess the independence between two binary random variables: the Pearson's χ^2 test and Fisher's exact test.

[2]In the discrete and exact case, $p_H\left(\mathcal{S}\right)$ can be conservative, as the condition

$$\Pr_{\mathcal{S}}\left(p_H\left(\mathcal{S}\right) \leq \alpha \mid H\right) \leq \alpha$$

usually holds with the rightmost inequality $\leq \alpha$ as a strict inequality $< \alpha$ for most of the values of α, and $p_H\left(\mathcal{S}\right)$ is said to be *stochastically dominated* by the Uniform Distribution.

Variables	$\mathcal{P} \subseteq t$	$\mathcal{P} \not\subseteq t$	Row totals
$t \in \mathcal{S}^1$	$\mathsf{z}^{\mathcal{P}}_{\mathcal{S}^1}$	$n_1 - \mathsf{z}^{\mathcal{P}}_{\mathcal{S}^1}$	n_1
$t \in \mathcal{S}^0$	$\mathsf{z}^{\mathcal{P}}_{\mathcal{S}^0}$	$n_0 - \mathsf{z}^{\mathcal{P}}_{\mathcal{S}^0}$	n_0
Column totals	$\mathsf{z}^{\mathcal{P}}_{\mathcal{S}}$	$n - \mathsf{z}^{\mathcal{P}}_{\mathcal{S}}$	n

Table 2.1: 2×2 contingency table summarizing the appearance of the pattern $\mathcal{P}$ in $\mathcal{S}^0$ and $\mathcal{S}^1$.

Pearson's χ^2 test

One of the first proposed statistical test to assess the difference of frequencies of events between different sets of samples is Pearson's χ^2 test (Pearson, 1900).

Define the random variables $X^0_{\mathcal{P}}, X^1_{\mathcal{P}}, X^0_{\neg\mathcal{P}}, X^1_{\neg\mathcal{P}}$, where $X^j_{\mathcal{P}}$ models the number of samples (t, ℓ) where $\mathcal{P} \subseteq t$ and $\ell = \ell^j$; thus, $\mathsf{z}_{\mathcal{P}}(\mathcal{S}^j)$ corresponds to the observed value of $X^j_{\mathcal{P}}$. Instead, $X^j_{\neg\mathcal{P}}$ models the number of samples (t, ℓ) where $\mathcal{P} \not\subseteq t$ and $\ell = \ell^j$. We may observe that such random variables correspond to the 4 innermost entries of the contingency table shown in Table 2.1.

Then, define the test statistic $T_{H_{\mathcal{P}}}$ for $\mathcal{P}$ as

$$T_{H_{\mathcal{P}}} = \sum_{i \in \{\mathcal{P}, \neg\mathcal{P}\},\, j \in \{0,1\}} \frac{\left(X^j_i - \mathbb{E}\left[X^j_i\right]\right)^2}{\mathbb{E}\left[X^j_i\right]} \ .$$

An important property of $T_{H_{\mathcal{P}}}$ is that it converges in distribution to a *chi-squared* random variable χ^2 with 1 degree of freedom for $n \to +\infty$. This observation follows from observing that all X^j_i converge to normal distributions from the Central Limit Theorem.

Let the observed value of $T_{H_{\mathcal{P}}}$ be $\hat{T}_{H_{\mathcal{P}}}(\mathcal{S})$ where, under the null hypothesis $H_{\mathcal{P}} =$ " $\mathcal{P}$ is not associated to the labels", we have that $\mathbb{E}\left[X^j_i\right]$ can be estimated from the marginals of the (observed) contingency table:

$$\mathbb{E}\left[X^j_{\mathcal{P}}\right] = \frac{n_j \mathsf{z}^{\mathcal{P}}_{\mathcal{S}}}{n} \ , \ \mathbb{E}\left[X^j_{\neg\mathcal{P}}\right] = \frac{n_j \left(n - \mathsf{z}^{\mathcal{P}}_{\mathcal{S}}\right)}{n} \ .$$

Since $T_{H_{\mathcal{P}}} \to \chi^2$, the p-value $p^{\chi^2}_{H_{\mathcal{P}}}(\mathcal{S})$ can be computed from the tails of χ^2 distribution; it is defined as

$$p^{\chi^2}_{H_{\mathcal{P}}}(\mathcal{S}) = \Pr\left(\chi^2 \geq \hat{T}_{H_{\mathcal{P}}}(\mathcal{S})\right) \ .$$

As Pearson's Chi-squared test relies on the asymptotic convergence of $T_{H_{\mathcal{P}}}$ as $n \to +\infty$, it is denoted as an *asymptotic test*.

Fisher's exact test

One of the most employed statistical test to assess the association of two categorical random variable is Fisher's exact test (Fisher, 1922). If differs from Pearson's χ^2 test by the fact that it is an *exact test*, as it considers the exact distribution of the test statistics, and not its asymptotic distribution.

Fisher's exact test is a *conditional* test: it assumes all the *marginals* $(\mathsf{z}_{\mathcal{S}}^{\mathcal{P}}, n_1, n)$ of the contingency table for every pattern $\mathcal{P}$ to be fixed by design of the experiment; under the *null hypothesis* $H_{\mathcal{P}}$ of independence between variables $\mathbb{1}[\mathcal{P} \subseteq t]$ and $\mathbb{1}[\ell = \ell^1]$, the number $X_{\mathcal{P}}^1$ of samples with label ℓ^1 containing $\mathcal{P}$ follows a hypergeometric distribution:

$$\Pr\left(X_{\mathcal{P}}^1 = a \mid \mathsf{z}_{\mathcal{S}}^{\mathcal{P}}, n_1, n\right) = \binom{n_1}{a}\binom{n - n_1}{\mathsf{z}_{\mathcal{S}}^{\mathcal{P}} - a} \Big/ \binom{n}{\mathsf{z}_{\mathcal{S}}^{\mathcal{P}}} = \mathsf{h}(a, \mathcal{P}, \mathcal{S}) \quad . \tag{2.2}$$

The p-value $p_{H_{\mathcal{P}}}^{\mathsf{F}}(\mathcal{S})$ by Fisher's test, under the null hypothesis $H_{\mathcal{P}}$ for the pattern $\mathcal{P}$, is computed by summing all the probabilities $\mathsf{h}(a, \mathcal{P}, \mathcal{S})$ that are smaller than $\mathsf{h}\left(\mathsf{z}_{\mathcal{S}^1}^{\mathcal{P}}, \mathcal{P}, \mathcal{S}\right)$:

$$p_{H_{\mathcal{P}}}^{\mathsf{F}}(\mathcal{S}) = \sum_{a:\mathsf{h}(a,\mathcal{P},\mathcal{S}) \leq \mathsf{h}\left(\mathsf{z}_{\mathcal{S}^1}^{\mathcal{P}}, \mathcal{P}, \mathcal{S}\right)} \mathsf{h}(a, \mathcal{P}, \mathcal{S}) \quad .^3 \tag{2.3}$$

Other tests, such as Barnard's exact test (Barnard, 1945), rely on a less restrictive set of assumptions on the data generative process than Fisher's test, at the cost of increased complexity in the definition of the p-value. In Chapter 4 we will discuss in more detail such test, and how it is possible to integrate it effectively into the setting of Significant Pattern Mining.

2.2.2 Multiple Hypothesis Testing

As we discussed in Section 2.2.1, testing a single hypothesis H can be done by comparing its p-value $p_H(\mathcal{S})$ to a fixed threshold α; rejecting H when $p_H(\mathcal{S}) \leq \alpha$ guarantees that the probability of making a *false discovery* (e.g., rejecting an hypothesis when it is a true null hypothesis) is not larger than α. However, when multiple hypotheses are considered, the critical issue of the *Multiple Hypothesis Testing Problem* arises. In fact, when we test m hypotheses against a fixed significance threshold α, we expect αm of them to have p-values below α, even if all hypotheses are true null hypotheses.

More formally, let $\mathcal{H} = \{H_1, \ldots, H_m\}$ be a set of m hypotheses. Let $\mathsf{NH}(\mathcal{H})$ be the subset of true null hypotheses of $\mathcal{H}$, and $\mathsf{R}(\mathcal{H}, \mathcal{A}, \mathcal{S})$ be the subset of the hypotheses of $\mathcal{H}$ that, given a procedure $\mathcal{A}$, are rejected by $\mathcal{A}$ using $\mathcal{S}$. If $\mathcal{A}$ is

[3]This definition of the test is "two-sided", as both ranges $X_{\mathcal{P}}^1 \geq \mathsf{z}_{\mathcal{S}^1}^{\mathcal{P}}$ and $X_{\mathcal{P}}^1 \leq \mathsf{z}_{\mathcal{S}^1}^{\mathcal{P}}$ are considered to seek "more extreme" outcomes. Instead, a "one-sided" test would consider only one side of the range for $X_{\mathcal{P}}^1$.

"reject H if $p_H(\mathcal{S}) \leq \alpha$", we have

$$\mathsf{R}(\mathcal{H}, \mathcal{A}, \mathcal{S}) = \{H : p_H(\mathcal{S}) \leq \alpha, H \in \mathcal{H}\} \ .$$

Assume, for now, that all $H \in \mathcal{H}$ are true null hypotheses, that is $\mathsf{NH}(\mathcal{H}) = \mathcal{H}$. Then, it holds by linearity of expectation that

$$\mathbb{E}_{\mathcal{S}}[|\mathsf{R}(\mathcal{H}, \mathcal{A}, \mathcal{S})|] = \alpha m \ .$$

Furthermore, this holds for any existing dependence structure between the hypotheses. This problem naturally arises in Significant Pattern Mining and, more generally, in Knowledge Discovery, since one is interested in evaluating the significance of a large number of complex patterns that may appear in large datasets. For example, the language $\mathfrak{L}$ for itemsets is composed by all non-empty subsets of an alphabet $\mathcal{I}$; thus, we would have $m = |\mathfrak{L}| = 2^{|\mathcal{I}|} - 1$. In many settings, reporting false discoveries is very expensive; in biology, for example, Knowledge Discovery methods are often of exploratory nature, guiding successive follow-up experiments to validate the reported interesting patterns. Thus, a large number of spurious discoveries may lead to extremely expensive consequences; it is of critical importance to provide rigorous guarantees on the statistical quality of the set of discovered patterns.

To do so, one has to modify the criteria "$p_H(\mathcal{S}) \leq \alpha$" used to reject hypotheses with something potentially more restrictive; at the same time, it is important to not sacrifice the overall *power* of the procedure, that is, without incurring in an unacceptable rate of false negatives.

2.2.3 Controlling the Family-Wise Error Rate ($FWER$)

One common approach to address the Multiple Hypothesis Testing Problem is to design procedures that control the *Family-Wise Error Rate (FWER)*. This metric is defined as the probability of reporting at least one false positive. More precisely, recall the definitions of $\mathcal{H}$ as a set of m hypotheses $\mathcal{H} = \{H_1, \ldots, H_m\}$, $\mathsf{NH}(\mathcal{H})$ the subset of true null hypotheses of $\mathcal{H}$, and $\mathsf{R}(\mathcal{H}, \mathcal{A}, \mathcal{S})$ the subset of the hypotheses of $\mathcal{H}$ that, given a procedure $\mathcal{A}$, are rejected by $\mathcal{A}$ on $\mathcal{S}$. The $FWER$ of the procedure $\mathcal{A}$ is then defined as

$$FWER = \Pr_{\mathcal{S}}(|\mathsf{NH}(\mathcal{H}) \cap \mathsf{R}(\mathcal{H}, \mathcal{A}, \mathcal{S})| > 0) \ .$$

The goal in the design of $\mathcal{A}$ is to maximize the number $|\mathsf{R}(\mathcal{H}, \mathcal{A}, \mathcal{S})|$ of discoveries, that is, the number of rejected hypotheses, while keeping $FWER \leq \alpha$, for a given α set by the user.

The Bonferroni and Holm corrections

A standard method to bound the $FWER$ below α is the *Bonferroni correction* (Bonferroni, 1936). Such procedure $\mathcal{A}^{\mathsf{B}}$ is based on comparing the p-values of all the m hypotheses of $\mathcal{H}$ with the threshold α/m; therefore, $\mathcal{A}^{\mathsf{B}}$ is simply

$$\text{reject } H \text{ if } p_H(\mathcal{S}) \leq \frac{\alpha}{m} \ .$$

This guarantees $FWER \leq \alpha$ from a union bound over m events. However, since the number m of hypotheses can be huge, such as in many Significant Pattern Mining tasks, this approach results in limited statistical power: it is generally very difficult to reject hypotheses using such small threshold (Webb, 2006, 2007, 2008). A more refined solution than the Bonferroni correction was proposed by Holm (1979), often referred to as the Bonferroni-Holm correction, to reject hypotheses bounding the $FWER$ below α. While the Bonferroni-Holm correction is uniformly more powerful than $\mathcal{A}^{\mathsf{B}}$, as the set of its rejected hypotheses is always a superset of the set rejected by $\mathcal{A}^{\mathsf{B}}$, it still requires the p-values to be very small, at least proportional to α/m (and at least one below α/m), and thus usually do not provide significant advantages when m is very large.

Tarone's correction

In this Section we discuss how, in certain conditions, it is possible to obtain procedures drastically more powerful the standard methods mentioned above.

In discrete settings, it is often the case that the p-value $p_H(\mathcal{S})$ for a given hypothesis H is lower bounded by a positive constant ψ_H, such that

$$p_H(\mathcal{S}) \geq \psi_H \ , \ \forall \mathcal{S} \ .$$

This typically happens because statistical tests are computed on a limited number of samples, and therefore "the evidence" for rejecting the null hypothesis cannot exceed a certain threshold. In fact, when this holds, we may observe that H will *never* be rejected when the employed significance threshold δ for H is $\delta < \psi_H$. A breakthrough observation, that goes back to John Tuckey, Gart et al. (1979), Mantel (1980), and Tarone (1990), is that such hypotheses, called *untestable*[4], do not have to be taken into account as potential false positives, as they have no chance to be rejected. Tarone (1990) proposed a modification of the Bonferroni correction that is capable of excluding from consideration untestable hypotheses, obtaining a much more permissive significance threshold $\alpha/m^\star$ for the remaining $m^\star$ hypothesis, resulting in improved statistical power. In particular, he remarked that when p-values $p_H(\mathcal{S})$ are computed using conditional tests, such as Fisher's exact test, then their corresponding lower

[4]The term *"untestable"* may be misleading. In fact, such hypothesis *can be tested*, but are *"not worth to be tested"*, since there is no chance that they are rejected.

bounds ψ_H are functions of the marginals of the observed contingency tables, and therefore are readily available. Most importantly, the marginals do not give information on the actual value of $p_H(\mathcal{S})$, but only on its range; this allows to avoid any potential "selection bias" in discarding untestable hypothesis.

We define Tarone's correction as follows. Let $\mathsf{k}(\delta)$ be the number of hypotheses of $\mathcal{H}$ with minimum attainable p-value $\psi_H \leq \delta$:

$$\mathsf{k}(\delta) = |\{H : \psi_H \leq \delta, H \in \mathcal{H}\}| \ .$$

Tarone's correction procedure $\mathcal{A}^{\mathsf{T}}$ is then

$$\text{Let } \delta^* = \max\left\{\delta : \delta \leq \frac{\alpha}{\mathsf{k}(\delta)}\right\} \ . \text{ Reject all } H : p_H(\mathcal{S}) \leq \delta^* \ .$$

In many cases, $\delta^* \gg \alpha/m$, resulting in large gains in terms of statistical power w.r.t. Bonferroni and Bonferroni-Holm corrections.

A breakthrough method in Significant Pattern Mining is the work by Terada et al. (2013a), that proposes LAMP, the first method to identify Significant Patterns using Tarone's procedure $\mathcal{A}^{\mathsf{T}}$. An equivalent, but computationally more efficient strategy, was later proposed by Minato et al. (2014). We will describe in more detail such methods in Section 2.3, and in Chapter 3 and Chapter 4.

Resampling and Permutation Testing

As we discussed, the procedure of Tarone often yields improved statistical power than the "standard" Bonferroni and Bonferroni-Holm corrections. Nevertheless, the resulting significance threshold $\delta^\star$ can still be fairly conservative, as it ignores correlations between the hypothesis of $\mathcal{H}$. In fact, when the corresponding test statistics are correlated, the *effective* number of hypothesis that may result in a false discovery may be substantially smaller than $\mathsf{k}(\delta^\star)$. This scenario is typical in Significant Pattern Mining, since patterns are usually strongly correlated.

A solution to this issue is to handle the joint distribution of the test statistics of all subsets of the hypothesis; this is often impossible in practice, due to the complex nature of the hypotheses one is interested to test, and because such joint distributions may not be available in closed form. An alternative solution is to rely on resampling-based strategies to estimate such joint distributions.

One of such strategies has been proposed by Westfall and Young (1993) and it is denoted *"Westfall-Young (WY) permutation testing"*. The idea is to apply to the available data $\mathcal{S}$ a transformation $g(\cdot)$ from a space of transformations G, such that all the test statistics of $\mathcal{H}$ are distributed as null hypotheses. One example of G that applies to the setting we introduced in Section 2.1 is the set of all possible permutations of the labels of the samples of $\mathcal{S}$, where $g(\mathcal{S})$ is a specific permutation $g \in \mathsf{G}$ applied to $\mathcal{S}$. Intuitively, we should be able to understand which hypotheses

should be rejected by comparing the values of the test statistics computed on the original data $\mathcal{S}$ with the test statistics computed on the resampled data $g(\mathcal{S})$. More precisely, let $q_{\mathcal{H}}^{\mathsf{WY}}(\mathsf{G}, \mathcal{S}, \alpha)$ be defined as the α-quantile of the distribution of the maximum test statistic of $\mathcal{H}$, computed over the transformations of G:

$$q_{\mathcal{H}}^{\mathsf{WY}}(\mathsf{G}, \mathcal{S}, \alpha) = \min\left\{x : \frac{1}{|\mathsf{G}|}\sum_{g\in\mathsf{G}} \mathbb{1}\left[\max\left\{\hat{T}_H(g(\mathcal{S})) : H \in \mathcal{H}\right\} \geq x\right] \leq \alpha\right\} \ .$$

Naturally, the computation of $q_{\mathcal{H}}^{\mathsf{WY}}(\mathsf{G}, \mathcal{S}, \alpha)$ is often impossible as $|\mathsf{G}|$ may be impractically large (i.e., for permutations it is $|\mathsf{G}| = n!$); still, one can sample uniformly at random m elements $\mathcal{G}$ from G and evaluate the empirical version $q_{\mathcal{H}}^{\mathsf{WY}}(\mathcal{G}, \mathcal{S}, \alpha)$ of $q_{\mathcal{H}}^{\mathsf{WY}}(\mathsf{G}, \mathcal{S}, \alpha)$. As $m = |\mathcal{G}|$ grows, the estimate $q_{\mathcal{H}}^{\mathsf{WY}}(\mathcal{G}, \mathcal{S}, \alpha)$ converges to $q_{\mathcal{H}}^{\mathsf{WY}}(\mathsf{G}, \mathcal{S}, \alpha)$; in practice, values of m in the range $[10^3, 10^4]$ are usually sufficient to get accurate estimations.

The WY procedure $\mathcal{A}^{\mathsf{WY}}$ is defined as follows:

$$\text{Sample } \mathcal{G}, \text{ and compute } \tilde{T} = q_{\mathcal{H}}^{\mathsf{WY}}(\mathcal{G}, \mathcal{S}, \alpha). \text{ Reject all } H : \hat{T}_H(\mathcal{S}) \geq \tilde{T}.$$

We remark that an equivalent formulation for $q_{\mathcal{H}}^{\mathsf{WY}}(\mathcal{G}, \mathcal{S}, \alpha)$ that we will use is

$$q_{\mathcal{H}}^{\mathsf{WY}}(\mathcal{G}, \mathcal{S}, \alpha) = \max\left\{x : \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}} \mathbb{1}\left[\min\left\{p_H(g(\mathcal{S})) : H \in \mathcal{H}\right\} \leq x\right] \leq \alpha\right\} \ ,$$

where the α-quantile of p-values is of interest. In this case, $\mathcal{A}^{\mathsf{WY}}$ is

$$\text{Sample } \mathcal{G}, \text{ and compute } \tilde{\delta} = q_{\mathcal{H}}^{\mathsf{WY}}(\mathcal{G}, \mathcal{S}, \alpha). \text{ Reject all } H : p_H(\mathcal{S}) \leq \tilde{\delta}.$$

It is simple to prove that $\mathcal{A}^{\mathsf{WY}}$ controls the $FWER$ at level α in the weak-sense, that is, when all hypotheses of $\mathcal{H}$ are null hypotheses $\mathcal{H} = \mathsf{NH}(\mathcal{H})$, as $q_{\mathcal{H}}^{\mathsf{WY}}(\mathcal{G}, \mathcal{S}, \alpha)$ directly estimate the highest significance threshold to have $FWER \leq \alpha$. $\mathcal{A}^{\mathsf{WY}}$ controls the $FWER$ at level α in the strong sense (for all $\mathsf{NH}(\mathcal{H}) \subseteq \mathcal{H}$) when a sufficient, yet not necessary (Romano and Wolf, 2005; Westfall and Troendle, 2008), condition denoted *subset pivotality* (Westfall and Young, 1993) is verified. Such technical condition[5] is usually not easy to verify in practice, but holds in many situations. Interestingly, WY permutation testing is not only very powerful and useful in practice but it has been shown to be asymptotically optimal (Meinshausen et al., 2011).

[5]Subset pivotality holds (Westfall and Troendle, 2008) when the distributions of

$$\begin{aligned} &\{\max\{T_H : H \in K\} \mid \mathsf{NH}(K) = K\} \ , \\ \text{and } &\{\max\{T_H : H \in K\} \mid \mathsf{NH}(\mathcal{H}) = \mathcal{H}\} \end{aligned}$$

are identical for all $K \subseteq \mathcal{H}$; this implies that the transformations that make all hypotheses nulls make $q_{\mathcal{H}}^{\mathsf{WY}}(\mathcal{G}, \mathcal{S}, \alpha)$ a consistent estimator of (an upper bound to) the $FWER$.

An efficient application of the WY procedure $\mathcal{A}^{\mathsf{WY}}$ is not straightforward; in fact, it requires to compute $q_{\mathcal{H}}^{\mathsf{WY}}(\mathcal{G}, \mathcal{S}, \alpha)$ over a large set $\{g(\mathcal{S}) : g \in \mathcal{G}\}$ of resampled datasets and, for each one of them, to solve an optimization over the hypothesis $\mathcal{H}$; when $\mathcal{H}$ is large and complex, as in Significant Pattern Mining, it is not possible to compute $\max_{H \in \mathcal{H}} \left\{\hat{T}_H(\mathcal{S})\right\}$ by enumerating $\mathcal{H}$ exhaustively, and smarter strategies have to be devised. We will discuss such ideas in Section 2.3.

2.2.4 Controlling other Error Rates

As we discussed in the previous Section, rigorous guarantees on the discovery of false positives is critical in many applications. While in many settings it is crucial to avoid *any* false discovery by controlling the $FWER$, in other cases it may be preferable to tolerate a larger, but still *controlled*, number of false discoveries if this leads to an higher number of overall discoveries and, consequently, to higher power and a lower false negative rate. To this aim, more permissive error rates have been proposed; we briefly introduce them here, and postpone formal definitions to later Sections.

The *Generalized Family-Wise Error Rate* (g-$FWER$) (Lehmann and Romano, 2012) is an extension of the $FWER$, whose aim is to bound the probability that at least g false discoveries are in the output; when $g = 1$, the g-$FWER$ reduces to the $FWER$. Another fundamental measure is the False Discovery Rate (FDR) (Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001), the *expected proportion* of false discoveries over the set of overall discoveries. An alternative to the FDR is the *False Discovery Proportion (FDP)* (Romano et al., 2006a; Lehmann and Romano, 2012; Romano et al., 2006b), defined as the probability of rejecting a set of hypothesis with a fraction of false discoveries higher than ζ, for some $\zeta \in [0, 1]$. While the g-$FWER$ requires to fix g a priori, that may not be simple to do, a bound on the FDR (or the FDP) gives more flexibility as it adapts to the number of overall discoveries: it is more permissive (allowing more false discoveries) when many hypothesis can be rejected, and more restrictive otherwise.

Controlling such more permissive Error Rates in Significant Pattern Mining is a challenge not yet addressed by available methods. We discuss in Chapter 3 how a simple modification of WY permutation testing yields an efficient and powerful procedure that controls the g-$FWER$ and the FDP.

2.3 Significant Pattern Mining

LAMP: Tarone's correction for Significant Pattern Mining

LAMP (Terada et al., 2013a) is the first method that combined the correction of Tarone (1990) in the context of Multiple Hypothesis Testing for Significant Pattern Mining. In particular, it exploits the following observations.

For a pattern $\mathcal{P}$, recall from Section 2.1.1 the definitions of the support $\mathsf{z}^{\mathcal{P}}_{\mathcal{S}^1}$ in the set of samples $\mathcal{S}^1$ and the support $\mathsf{z}^{\mathcal{P}}_{\mathcal{S}}$ in all samples $\mathcal{S}$. Here we remark that the p-value $p^{\mathsf{F}}_{H_{\mathcal{P}}}(\mathcal{S})$ for $\mathcal{P}$ from Fisher's exact test, defined in Section 2.2.1, can be expressed as a function $p_{\mathcal{P}}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{S}^1}\right)$ of $\mathsf{z}^{\mathcal{P}}_{\mathcal{S}^1}$ only, as the marginals $\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{S}}, n_1, n\right)$ are fixed by design:

$$p^{\mathsf{F}}_{H_{\mathcal{P}}}(\mathcal{S}) = \sum_{a:\mathsf{h}(a,\mathcal{P},\mathcal{S})\leq \mathsf{h}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{S}^1},\mathcal{P},\mathcal{S}\right)} \mathsf{h}(a,\mathcal{P},\mathcal{S}) = p_{\mathcal{P}}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{S}^1}\right) \quad . \tag{2.4}$$

The fact that marginals are fixed implies that the domain of $p_{\mathcal{P}}(\cdot)$ is *finite*, since

$$\begin{aligned}
&\mathsf{z}^{\mathcal{P}}_{\mathcal{S}^1} \in \left[\check{\mathsf{z}}^{\mathcal{P}}_{\mathcal{S}^1}, \hat{\mathsf{z}}^{\mathcal{P}}_{\mathcal{S}^1}\right], \text{ with}\\
&\check{\mathsf{z}}^{\mathcal{P}}_{\mathcal{S}^1} = \max\left\{0, n_1 - \left(n - \mathsf{z}^{\mathcal{P}}_{\mathcal{S}}\right)\right\},\\
&\hat{\mathsf{z}}^{\mathcal{P}}_{\mathcal{S}^1} = \min\left\{n_1, \mathsf{z}^{\mathcal{P}}_{\mathcal{S}}\right\}.
\end{aligned}$$

This observations implies the existence of a *minimum attainable p-value* $\psi\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{S}}\right)$, which depends on $\mathsf{z}^{\mathcal{P}}_{\mathcal{S}}$ only, as n_1, and n are fixed for all patterns. $\psi\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{S}}\right)$ is computed considering the most biased cases in equation 2.4:

$$\psi\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{S}}\right) = \min\left\{p_{\mathcal{P}}\left(\check{\mathsf{z}}^{\mathcal{P}}_{\mathcal{S}^1}\right), p_{\mathcal{P}}\left(\hat{\mathsf{z}}^{\mathcal{P}}_{\mathcal{S}^1}\right)\right\} \quad .$$

For a significance threshold δ, we denote the set of *testable patterns* $\mathsf{T}(\delta)$ w.r.t. δ as

$$\mathsf{T}(\delta) = \left\{\mathcal{P} : \psi\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{S}}\right) \leq \delta, \mathcal{P} \in \mathfrak{L}\right\} \quad .$$

as the set of patterns that have a chance of being significant when their p-values is compared to δ. The correction procedure of Tarone in the context of Significant Pattern Mining can be adapted, resulting in the following procedure:

$$\text{Let } \delta^* = \max\left\{\delta : \delta \leq \frac{\alpha}{|\mathsf{T}(\delta)|}\right\} \text{ . Output all } \mathcal{P} \in \mathsf{T}(\delta^*) : p_{\mathcal{P}}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{S}^1}\right) \leq \delta^* \quad .$$

However, computing $|\mathsf{T}(\delta)|$ is not straightforward as $\psi\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{S}}\right)$ is not antimonotonic in $\mathsf{z}^{\mathcal{P}}_{\mathcal{S}}$, and therefore it does not enjoy the computational advantages of methods designed for Frequent Pattern Mining we discussed in Section 2.1.2. To address this issue, the authors of LAMP noted that $\psi(x)$ is non-increasing for $x < n_1$ and minimized at $x = n_1$; thus, they defined the monotone version $\hat{\psi}(x)$ (non-increasing for all $x \in [1, n]$) of $\psi(x)$ as follows:

$$\hat{\psi}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{S}}\right) = \begin{cases} \psi\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{S}}\right) & \text{, if } \mathsf{z}^{\mathcal{P}}_{\mathcal{S}} \leq n_1,\\ \psi(n_1) & \text{, otherwise.} \end{cases}$$

This allows to define the set of *testable patterns* $\hat{\mathsf{T}}(\delta)$ w.r.t. δ and $\hat{\psi}(\cdot)$ as

$$\hat{\mathsf{T}}(\delta) = \left\{\mathcal{P} : \hat{\psi}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{S}}\right) \leq \delta, \mathcal{P} \in \mathfrak{L}\right\} \supseteq \mathsf{T}(\delta) \quad .$$

If we denote θ_δ as

$$\theta_\delta = \min\left\{\frac{x}{n} : \hat{\psi}(x) \leq \delta\right\} \quad ,$$

then, it holds

$$\hat{\mathsf{T}}(\delta) = \mathsf{FP}(\mathfrak{L}, \mathcal{S}, \theta_\delta) \quad .$$

The equivalence of $\hat{\mathsf{T}}(\delta)$ with the set of Frequent Patterns $\mathsf{FP}(\mathfrak{L}, \mathcal{S}, \theta_\delta)$ is the key that allowed LAMP to combine Tarone's correction with existing efficient and optimized methods for Frequent Pattern Mining. The resulting LAMP correction to bound the $FWER$ below α is:

Let $\delta^* = \max\left\{\delta : \delta \leq \dfrac{\alpha}{|\mathsf{FP}(\mathfrak{L}, \mathcal{S}, \theta_\delta)|}\right\}$. Output all $\mathcal{P} \in \mathsf{FP}(\mathfrak{L}, \mathcal{S}, \theta_{\delta^*}) : p_{\mathcal{P}}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{S}^1}\right) \leq \delta^*$.

The computation of δ^* was performed by guessing multiple values of θ_δ in the first version of LAMP (Terada et al., 2013a) and computing $|\mathsf{FP}(\mathfrak{L}, \mathcal{S}, \theta_\delta)|$ for each guess; since repeating this operation is expensive, a more refined approach, that performs only *one* depth first branch-and-bound enumeration of $\mathfrak{L}$ was proposed by Minato et al. (2014).

As we will discuss in the next Section, the monotone lower bound function $\hat{\psi}(\cdot)$ to p-values is also useful for embedding the Westfall-Young (WY) permutation procedure in Significant Pattern Mining.

WY permutation testing for Significant Pattern Mining

In this Section we present the state-of-the-art methods to perform the WY correction procedure for Multiple Hypothesis Testing in Significant Pattern Mining.

Denote with $\mathcal{G}$ a set of m permutations of the labels of $\mathcal{S}$, wher $g(\mathcal{S})$ is a permuted sample with $g \in \mathcal{G}$. To compute the significance threshold $\delta^\star$, it is required to evaluate, for each $g \in \mathcal{G}$, the minimum p-value p^g over all patterns $\mathcal{P} \in \mathfrak{L}$ computed on $g(\mathcal{S})$, defined as

$$p^g = \min\left\{p^{\mathsf{F}}_{H_{\mathcal{P}}}(g(\mathcal{S})) : \mathcal{P} \in \mathfrak{L}\right\} \quad .$$

Then, $\delta^\star$ corresponds to the α-quantile of the set $\{p^g : g \in \mathcal{G}\}$.

The first method that efficiently applied the WY permutation testing procedure to Significant Pattern Mining is FastWY (Terada et al., 2013b), later refined by Terada et al. (2015). The idea of FastWY is to compute each minimum p-value p^g separately, avoiding to enumerate *all* patterns $\mathcal{P} \in \mathfrak{L}$. To do so, FastWY explores $\mathfrak{L}$ in a branch-and-bound fashion; let $\tilde{p}$ be the value of p^g that is known at a given point of the enumeration, with $\tilde{p} \geq p^g$. FastWY uses the function $\hat{\psi}(\cdot)$ to *prune* portions of

$\mathfrak{L}$ containing patterns that are not enough frequent in $g(\mathcal{S})$ to yield p-values smaller than $\tilde{p}$. Denote a pattern $\mathcal{P}_1$ and a child pattern $\mathcal{P}_2$ of $\mathcal{P}_1$, with, by definition, $\mathsf{z}_{\mathcal{S}}^{\mathcal{P}_1} \geq \mathsf{z}_{\mathcal{S}}^{\mathcal{P}_2}$. We may observe that

$$\hat{\psi}\left(\mathsf{z}_{\mathcal{S}}^{\mathcal{P}_1}\right) > \tilde{p} \implies \hat{\psi}\left(\mathsf{z}_{\mathcal{S}}^{\mathcal{P}_2}\right) \geq \hat{\psi}\left(\mathsf{z}_{\mathcal{S}}^{\mathcal{P}_1}\right) > \tilde{p} \geq p^g \ .$$

This means that if $\mathcal{P}_1$ is untestable w.r.t. $\tilde{p}$, because $\hat{\psi}(\mathsf{z}_{\mathcal{S}}^{\mathcal{P}_1}) > \tilde{p}$, then any child $\mathcal{P}_2$ of $\mathcal{P}_1$ is untestable, since $\hat{\psi}(\mathsf{z}_{\mathcal{S}}^{\mathcal{P}_2}) > \tilde{p}$, and therefore can be excluded from the search of p^g.

While drastically more efficient than a "naïve" implementation of the WY procedure, FastWY may still be computationally demanding in some cases. A method called WYlight (Llinares-López et al., 2015) was proposed to address this issue. WYlight focuses on computing exactly only the α-quantile of the set of minimum p-values $\{p^g : g \in \mathcal{G}\}$: Llinares-López et al. (2015) acknowledged that only the α-quantile is really needed to test the significance of the patterns with guaranteed $FWER \leq \alpha$, while the other minimum p-values can be ignored, as they are more expensive to compute. For this reason, WYlight has been shown to be more efficient than FastWY both in terms of running time and memory.

In addition to the contributions for Significant Pattern Mining mentioned above, recent work has extended the extraction of Statistically-sound Patterns (Hämäläinen and Webb, 2019) in other directions, for example searching for statistical dependency rules between itemsets and items (Hämäläinen, 2012), or using an holdout approach (Webb, 2007) and layered critical values (Webb, 2008) for correcting for Multiple Hypothesis Testing.

2.4 Statistical Learning Theory

As we discussed, Pattern Mining is a task of Knowledge Discovery and Data Mining that aims at detecting meaningful patterns in data. In many situations, the true goal of the analysis is not to discover interesting structures of the data itself, but rather to study the *underlying*, and often *unknown*, *process* that generated it: our aim is to understand some of its characteristics such that, possibly, we may be able to say something meaningful about future observations.

When describing Significant Pattern Mining, we implicitly followed this direction, in the sense that our main goal was to provide probabilistic guarantees, under proper assumptions, that the discovered patterns were truly significant with sufficient confidence w.r.t. to the underlying distribution, and not on merely interesting on the data by itself.

Statistical Learning Theory is an important branch of the theoretical foundations of Machine Learning that aims at providing quantitative probabilistic guarantees on the performances of learning algorithms. As we will discuss in the next Section, key

concepts of Statistical Learning Theory are of central importance in many applications, and are fundamental to study generalization properties of learning methods.

Analysis of random samples for Approximate Pattern Mining

In this work we are interested in the connection between concepts of Statistical Learning Theory and probabilistic bounds on the performance of randomized algorithms for the analysis of *samples* for Pattern Mining. We will assume that the data $\mathcal{S}$ is a *sample* from an unknown probability distribution μ. From the analysis of $\mathcal{S}$, our goal is to derive guaranteed conclusions about properties of μ. We are therefore interested in the study of the *sample complexity* of Pattern Mining tasks, defined as the relationship between the size of the sample and the obtainable accuracy of the analysis performed on it. As we discussed previously, there are two meanings of "sample" and μ in this context; they can, as we argue, be treated in an unified way.

The first meaning is *sample* as a *small random sample of a large dataset*: since mining patterns becomes more expensive as the dataset grows, it is reasonable to mine only a small random sample that fits into the main memory of the machine. To obtain desirable probabilistic guarantees on the quality of the approximation, one must study the *trade-off* between the *size of the sample* and the *quality of the approximation*. We will present many recently proposed methods based on key concepts from Statistical Learning Theory, that obtained more favourable trade-offs than comparable methods based on other techniques.

The second meaning is *sample* as a *sample from an unknown data generating distribution*: the whole data is seen as a collection of samples from an unknown distribution, and the goal of mining patterns from the available dataset is to discover knowledge about the distribution. This area is known as *statistically-sound pattern discovery* (Hämäläinen and Webb, 2019), and, as we will see in the following Sections, there are many different flavors of it, such as Significant Pattern Mining, as we already discussed.

As we already discussed, these two meanings of "sample" and distribution are only apparently distinct.

2.4.1 Uniform Convergence

Before describing key concepts from Statistical Learning Theory are useful for Pattern Mining applications, we define a fundamental concept in learning theory: *uniform convergence*.

As in Section 2.1.1, we consider a set of real-valued functions $\mathcal{F}$ from a domain $\mathcal{X}$ to an interval $[a, b] \subset \mathbb{R}$. We denote a sample $\mathcal{S}$ from a probability distribution μ, with each $s \in \mathcal{S}$ obtained i.i.d. from μ. We denote the *average value* of a function

$f \in \mathcal{F}$ over samples $s \in \mathcal{S}$ as

$$\mathsf{a}_f(\mathcal{S}) = \frac{1}{n}\sum_{i=1}^{n} f(s_i) \quad .$$

As we saw from Section 2.1.1, functions $f \in \mathcal{F}$ can be used to encode some notion of *interest* for patterns of a language $\mathfrak{L}$, such as the *frequency* of a given pattern $\mathcal{P}$ in $\mathcal{S}$. Since $\mathcal{S}$ is a random sample from a distribution μ, another important quantity is the *expectation* of f taken w.r.t. $\mathcal{S}$:

$$\mathbb{E}_{\mathcal{S}\sim\mu}[f] = \mathbb{E}_{\mathcal{S}\sim\mu}[\mathsf{a}_f(\mathcal{S})] \quad ,$$

that can be interpreted as the "true" value of $\mathsf{a}_f(\mathcal{S})$ according to μ. A natural question is how well $\mathsf{a}_f(\mathcal{S})$ *approximates* its expectation $\mathbb{E}_{\mathcal{S}\sim\mu}[f]$. We already know, from classic asymptotic results (e.g., the Central Limit Theorem), that $\mathsf{a}_f(\mathcal{S})$ converges to $\mathbb{E}_{\mathcal{S}\sim\mu}[f]$ as $n = |\mathcal{S}|$ goes to infinity; we are, however, interested in quantifying the *rate of the convergence* with finite-sample bounds. To do so, we denote the *Supremum Deviation* $\mathsf{D}(\mathcal{F},\mathcal{S})$ as the maximum absolute difference between $\mathsf{a}_f(\mathcal{S})$ and its expectation $\mathbb{E}_{\mathcal{S}\sim\mu}[f]$, over all $f \in \mathcal{F}$:

$$\mathsf{D}(\mathcal{F},\mathcal{S}) = \sup_{f\in\mathcal{F}} |\mathsf{a}_f(\mathcal{S}) - \mathbb{E}_{\mathcal{S}\sim\mu}[f]| \quad .$$

We define *uniform convergence*, for a given $\mathcal{S}$ and ε, as having $\mathsf{D}(\mathcal{F},\mathcal{S}) \leq \varepsilon$; that is, all estimates $\mathsf{a}_f(\mathcal{S})$ are uniformly close to (i.e., within ε from) their true values $\mathbb{E}_{\mathcal{S}\sim\mu}[f]$. Equivalently, uniform convergence implies simultaneous bounds on the expected values of $f \in \mathcal{F}$ from their estimates. Our goal is to derive functions $\mathsf{d}(\mathcal{F},\mathcal{S},\delta)$ that upper bounds $\mathsf{D}(\mathcal{F},\mathcal{S})$ with confidence δ such that, with probability $\geq 1-\delta$ over $\mathcal{S}$,

$$\mathsf{D}(\mathcal{F},\mathcal{S}) \leq \mathsf{d}(\mathcal{F},\mathcal{S},\delta) \quad ,$$

possibly exploiting our knowledge of $\mathcal{F}$ and information obtainable from $\mathcal{S}$.

As we will see in Chapter 5, Chapter 6, and Chapter 7, probabilistic bounds on the largest error of the empirical averages are typically obtained by adding to the empirically estimated error a term that depends on the *complexity* of the functions. Both *distribution-free* concepts of complexity, and *distribution* and *data-dependent* complexities, have been proposed as breakthroughs with great success for this problem.

A distribution-free complexity captures the expressive power of functions, independently of the distribution that generates the data; for example, the *VC dimension* (Vapnik and Chervonenkis, 1971) is a combinatorial notion of complexity that measures the capacity of sets of binary functions to partition in all possible subsets a set of input points of a given size. We will define it more formally and use it to analyze an algorithm based on random sampling in Chapter 5.

As distribution-free complexities need to account for worst-cases instances, data-dependent complexities use, instead, the available data to directly measure the expressiveness of class of functions of interest; for this reason, such measures often provides sharper results. One of the most interesting notions of data-dependent measure of complexity of sets of functions is the Rademacher Complexity (Shalev-Shwartz and Ben-David, 2014; Mitzenmacher and Upfal, 2017), extensively studied in the context of classification and uniform convergence by Koltchinskii and Panchenko (2000), Bartlett et al. (2002), Bartlett and Mendelson (2002) and others. We formally define the Rademacher Complexity, and other related quantities, in Chapter 6 and Chapter 7.

2.5 Approximate Pattern Mining

The idea of mining a small random sample of a large dataset to speed up the pattern extraction step was proposed for the case of itemsets by Toivonen (1996) shortly after the first algorithm for the task had been introduced. The trade-off between the sample size and the quality of the approximation obtained from the sample has been progressively better characterized in successive works by Chakaravarthy et al. (2009) and Pietracaprina et al. (2010), with significant improvements by Riondato and Upfal (2014, 2015) due to the use of concepts from Statistical Learning Theory, such as the aforementioned VC dimension and Rademacher Complexity. Similar concepts were also applied to mining approximate interesting subgroups by Riondato and Vandin (2018), while Servan-Schreiber et al. (2018a) and Santoro et al. (2020) obtained bounds on the (empirical) VC dimension and Rademacher averages for sequential patterns.

In the context of Statistically-sound Pattern Mining (Hämäläinen and Webb, 2019), Kirsch et al. (2012) develop an algorithm to discover Significant Frequent Itemsets w.r.t. a probabilistic model, where each item is inserted independently at random in transactions; Riondato and Vandin (2014) introduce the problem of finding True Frequent Itemsets, i.e., the itemsets that are frequent w.r.t. the unknown distribution using empirical VC dimension. We discuss this specific application in more detail in Chapter 6, while in Chapter 4 we see how uniform convergence, obtained though bounds on the Rademacher Complexity, can be embedded in Significant Pattern Mining to discover significant patterns with bounded $FWER$.

Chapter 3

Efficient Mining of the Most Significant Patterns with Permutation Testing

3.1 Introduction

Significant Pattern Mining is an extension of Frequent Pattern Mining in which each transaction is assigned a binary class label and the goal is to identify patterns having *significant association* with one of the class labels. Significance is commonly assessed using a statistical test (e.g., Fisher's exact test, defined in Section 2.2.1), that provides a p-value quantifying the probability that the association observed in real data arises due to chance alone.

As we discussed in Section 2.2, one of the critical issues in Significant Pattern Mining is the *Multiple Hypothesis Testing Problem*, due to the huge number of patterns appearing in large datasets. Standard methods to correct for Multiple Hypothesis Testing by controlling the Family-Wise Error Rate ($FWER$), such as the Bonferroni correction (Section 2.2.3) are often too conservative for Significant Pattern Mining. To cope with this issue, we described, in Section 2.3, the breakthrough work by Terada et al. (2013a), that proposes LAMP, the first method to identify significant patterns based on the work by Tarone (1990) to discard large amounts of patterns that cannot reach statistical significance, called *untestable*. Subsequent work by Minato et al. (2014) has improved the search strategy employed by LAMP to identify testable patterns. Even so, such methods suffer from limited power, since they do not account for the dependencies between the patterns.

Recently, methods based on the more powerful Westfall-Young (WY) permutation procedure have been proposed (described in detail in Section 2.2), first by Terada et al. (2013b) with FastWY and then by Llinares-López et al. (2015) with Westfall-Young light (WYlight for short). These methods, presented in Section 2.3, mine several permuted datasets to identify a threshold such that all patterns with p-value below the threshold can be flagged as statistically significant while controlling the $FWER$. Such methods achieve a higher power than methods based on Bonferroni correction, including LAMP, and the state-of-the-art, WYlight, has proved to be more efficient than FastWY also in terms of runtime and memory.

However, the extraction of significant patterns from large datasets is still challenging, with three crucial issues that are not addressed by currently available methods. First, in several cases the dependency among patterns leads to a huge number of statistically significant patterns even after multiple hypothesis correction. A common approach (used, e.g., by Terada et al. (2013b) and Llinares-López et al. (2015)) to partially alleviate this problem is to consider only closed patterns (Han et al., 2007), discarding patterns with redundant information content in terms of appearance in the dataset and of association with the class label. Even with this restriction the number of significant patterns can be extremely large and when this happens one would like to focus on the most significant ones, without resorting to filtering strategies after the expensive extraction of all significant patterns has been performed. A second and related issue is that current methods work by first identifying the exact corrected threshold for statistical significance, and only subsequently mining the real dataset:

when the number of significant patterns is huge, one would like to focus on the most significant ones without the burden of computing the exact significance threshold. Third, all methods may need to process several untestable patterns to identify the correct threshold for significance, resulting in a extremely large running time in particular for datasets with many low frequency patterns. These issues make current methods impractical in many cases, as shown by our experimental evaluation.

3.1.1 Contributions

In this work we focus on the problem of mining the most statistically significant patterns while rigorously controlling the $FWER$ of the returned set of patterns. In particular, in analogy with frequent pattern mining approaches, we focus on extracting the Top-k Statistically Significant Patterns. This problem is more challenging than the extraction of Top-k Frequent Patterns, given that statistical significance does not enjoy the anti-monotonicity property w.r.t. to pattern frequency. In this regards, our contributions are:

- we formally define the problem of mining *Top-k Statistically Significant Patterns*. Our definition allows to properly control the size of the output set while providing guarantees on the $FWER$ of the output.

- we design a novel algorithm, called TopKWY, for the problem above, which provides guarantees on the $FWER$ by using the Westfall-Young permutation testing procedure. TopKWY adapts to the distribution of significant patterns: it reports *all* significant patterns when their number is small, while it outputs only the most significant patterns when the number of significant patterns is huge. TopKWY is based on a exploration strategy similar to the one used by TopKMiner (Pietracaprina and Vandin, 2007), an efficient algorithm to identify the top-k frequent patterns. We prove that the use of such strategy guarantees that, in contrast to previous approaches, TopKWY will never explore *untestable* patterns.

- we introduce several bounds to prune *untestable* patterns that improve over the bound introduced by LAMP and used in WYlight as well. We show that such bounds can be effectively used within the exploration strategy employed by TopKWY and that it provides a significant speed-up for real datasets.

- we present variants of TopKWY to mine the Top-k Significant Patterns with control of the Generalized Family-Wise Error Rate (g-$FWER$) or the False Discovery Proportion (FDP), which trade an increase in the size of the output with a (potentially) higher, but still controlled, number of false discoveries. These variants lead to an increase in the *statistical power* in situations where the number of results reported controlling the $FWER$ is low.

- we conduct an extensive experimental evaluation of the use of TopKWY to extract significant itemsets and subgraphs, showing that TopKWY allows the extraction of statistically significant patterns for large datasets while having reasonable memory requirements. Surprisingly, for many datasets TopKWY improves over the state-of-the-art even when it is used to find *all* statistically significant patterns.

3.2 Background and Problem Definition

3.2.1 Significant Pattern Mining

In this Section we refresh the notation introduced in Section 2.1.1. Let the dataset $\mathcal{D} = \{s_1, s_2, \ldots, s_n\}$ be a multiset of n samples, where each $s \in \mathcal{D}$ is an element from a domain $\mathcal{X}$. Each sample s is composed by a pair (t, ℓ) such that t is a *transaction* from its domain $\mathcal{T}$, and ℓ is a *binary label* $\ell \in \{\ell^0, \ell^1\}$. We denote by n_1 the number of samples with label ℓ^1, and, without loss of generality, we assume that n_1 is the *minority class*, i.e. $n_1 \leq n/2$. We define a *pattern* $\mathcal{P}$ as an element of a *language* $\mathfrak{L}$, and for each transaction t we define the binary function $f_{\mathcal{P}} : \mathcal{X} \to \{0, 1\}$ such that $f_{\mathcal{P}} = 1$ if $\mathcal{P}$ is *contained* in the transaction t of the corresponding sample $s = (t, \ell)$ and $f_{\mathcal{P}} = 0$ otherwise. For example: in the case of *itemsets*, both $\mathcal{T}$ and $\mathfrak{L}$ are given by the set of (non-empty) subsets of a universe of binary features $\mathcal{I}$; for *subgraphs*, $\mathcal{T}$ is the set of vertex-labelled graphs, while $\mathcal{P}$ is an element of $\mathfrak{L}$ with the constraint that $\mathcal{P}$ is connected. Given a pattern $\mathcal{P}$, we define its support $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}$ as the number of transactions containing $\mathcal{P}$, that is $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}} = \sum_{i=1}^{n} f_{\mathcal{P}}(s_i)$. We denote by $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}$ the number of labelled ℓ^1 transactions containing $\mathcal{P}$. In this work we assume that patterns enjoy the *anti-monotonicity property*, such that for any child pattern $\mathcal{P}'$ of $\mathcal{P}$ (i.e., $\mathcal{P} \subset \mathcal{P}'$) the support $\mathsf{z}^{\mathcal{P}'}_{\mathcal{D}}$ of $\mathcal{P}'$ is $\mathsf{z}^{\mathcal{P}'}_{\mathcal{D}} \leq \mathsf{z}^{\mathcal{P}}_{\mathcal{D}}$. This holds for itemsets, subgraphs, and many other kind of patterns.

The objective of *Significant Pattern Mining* is to find patterns with *significant statistical association* to one of the two labels. In order to quantify the statistical association, a rigorous *statistical test* is performed. The distribution of a pattern over the sets of samples with different labels is described by a 2×2 contingency table (see Table 2.1 in Section 2.2.1). To test the association of $\mathcal{P}$ w.r.t. the labels, we compute a p-value $p_{H_{\mathcal{P}}}(\mathcal{S})$ that, under the null hypothesis $H_{\mathcal{P}} =$ " $\mathcal{P}$ is not associated to the labels ", quantifies the probability that the observed association is only due to chance. Fisher's exact test (Fisher, 1922), defined in Section 2.2.1, is often used to compute such p-value. Here we remark that the p-value $p^{\mathsf{F}}_{H_{\mathcal{P}}}(\mathcal{S})$ for $\mathcal{P}$ from Fisher's exact test, defined in Section 2.2.1, can be expressed as a function $p_{\mathcal{P}}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}\right)$ of $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}$ only, as the marginals

$\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, n_1, n\right)$ are fixed by design:

$$p^{\mathsf{F}}_{H_{\mathcal{P}}}(\mathcal{S}) = \sum_{a:\mathsf{h}(a,\mathcal{P},\mathcal{S})\leq \mathsf{h}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1},\mathcal{P},\mathcal{S}\right)} \mathsf{h}(a,\mathcal{P},\mathcal{S}) = p_{\mathcal{P}}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}\right) \quad . \tag{3.1}$$

3.2.2 Multiple Hypothesis Testing

As we discussed in Section 2.2, when only one pattern $\mathcal{P}$ is tested, it can be flagged as *significant* when its p-value is smaller than a *significance threshold* α fixed *a priori*. This guarantees that the probability of a false discovery (i.e., reporting $\mathcal{P}$ as significant when it is not) is bounded by α. However, the expected number of false discoveries, for fixed α, linearly grows with the size of the set of tested hypotheses $\mathcal{H}$. Therefore, an appropriate *Multiple Hypothesis Testing correction* of the significance threshold needs to be performed in order to obtain rigorous guarantees in terms of the number of false associations reported in output.

One common approach is to perform a correction in order to bound the *Family-Wise Error Rate* ($FWER$), which is defined as the probability of reporting at least one false positive (see also Section 2.2.2). Many methods are available for controlling the $FWER$, such as the Bonferroni and Bonferroni-Holm corrections, and the LAMP method, based on Tarone's correction, all defined in Section 2.2.2. However, as we discussed, all such methods may be conservative as they do not take into account the dependencies between patterns.

To solve this issue, the Westfall-Young (WY) permutation testing procedure can be used. As we discussed in Section 2.2.3, the WY procedure directly estimate the significance threshold to use for control $FWER$ below α as the α-quantile of minimum p-values over a set of m resampled datasets. In our case, every resampled dataset is given by randomly permuting the labels of samples of $\mathcal{D}$. Let $\mathcal{G}$ be a set of m permutations $\mathcal{G} = \left\{g^{(j)}, j \in [1,m]\right\}$ sampled uniformly at random from the set of all possible permutations of n labels G, and let $g^{(j)}(\mathcal{D})$ be the resampled version of $\mathcal{D}$ according to $g^{(j)}$. Let the set of hypotheses $\mathcal{H} = \{H_{\mathcal{P}} : \mathcal{P} \in \mathfrak{L}\}$ corresponding to patterns $\mathcal{P} \in \mathfrak{L}$. Then, let $p^{(j)}_{\min}$ be the minimum p-value over all $\mathcal{P} \in \mathfrak{L}$ computed on $g^{(j)}(\mathcal{D})$:

$$p^{(j)}_{\min} = \min\left\{p^{\mathsf{F}}_{H_{\mathcal{P}}}\left(g^{(j)}(\mathcal{D})\right) : \mathcal{P} \in \mathfrak{L}\right\} \quad .$$

The corrected significance threshold $\delta(\alpha)$ to use, according to the WY correction procedure, to bounds the $FWER \leq \alpha$ is then given by $\delta(\alpha) = q^{\mathsf{WY}}_{\mathcal{H}}(\mathcal{G},\mathcal{D},\alpha)$, where

$$\delta(\alpha) = q^{\mathsf{WY}}_{\mathcal{H}}(\mathcal{G},\mathcal{D},\alpha) = \max\left\{x : \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}\left[p^{(j)}_{\min} \leq x\right] \leq \alpha\right\} \quad .$$

As we discussed, when m is sufficiently large (typically in the order of 10^3 or 10^4) the estimate is usually very accurate.

The WY method does not provide an efficient way of computing the set $\{p_{\min}^{(j)}\}_{j=1}^{m}$ of minimum p-values. Therefore, a naïve implementation requires to exhaustively test *all* the hypothesis on all the m permuted datasets. For significant pattern mining, this means exploring all the patterns appearing in a dataset; since this operation can require exponential time, it is even more challenging to repeat the entire process m times, once for every permuted dataset. Terada et al. (2013b) proposed the first efficient implementation, FastWY, of the WY procedure for significant pattern mining. The identification of $\delta(\alpha)$ is based on a decremental search scheme, which starts with processing the set of Frequent Patterns $\mathsf{FP}(\mathfrak{L}, \mathcal{D}, \theta)$ with minimum frequency threshold $\theta = 1$ and iteratively decrements θ until an appropriate condition, guaranteeing that all values $\{p_{\min}^{(j)}\}_{j=1}^{m}$ have been computed, is achieved. Such condition is based on the lower bound function $\hat{\psi} : [1, n] \to [0, 1]$ to p-values from Fisher's exact test, introduced in LAMP, that we described in Section 2.3; such function $\hat{\psi}(x)$ is non-increasing in $x \in [1, n]$, such that

$$\hat{\psi}\left(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}\right) \leq p_{H_{\mathcal{P}}}^{\mathsf{F}}(\mathcal{D}) \quad ,$$

for all possible labelling of $\mathcal{D}$ (with n_1 labels ℓ^1). A more recent method by Terada et al. (2015), HWY, exploits a more efficient mining strategy and parallel computing to accelerate FastWY.

Llinares-López et al. (2015) proposed WYlight to efficiently compute the optimal value $\delta(\alpha)$ of the corrected significance threshold for controlling $FWER \leq \alpha$. The main improvement of WYlight is to avoid the exact computation of all the elements of the set $\{p_{\min}^{(j)}\}_{j=1}^{m}$ and to only produce its exact lower α-quantile. This result is obtained by maintaining an estimate of the α-quantile that is only lowered through the mining process. WYlight performs a depth first exploration of the patterns' search tree (Han et al., 2000) in which each pattern has support less or equal than its parent, and performs only one pattern mining instance, testing one pattern at a time and computing its p-value on all the m permuted datasets at the same time. WYlight maintains a support threshold σ, initialized at 1, that is raised during the execution of the algorithm, pruning patterns whose p-values cannot be in the lower α-quantile of $\{p_{\min}^{(j)}\}_{j=1}^{m}$. This is achieved by using the lower bound $\hat{\psi}(\sigma)$ on the minimum obtainable p-value for patterns of support $\leq \sigma$, which allows to effectively prune the search tree. However, during the computation of $\delta(\alpha)$, some patterns with support $< \hat{\psi}^{-1}(\delta(\alpha))$ may be processed, due to the depth first procedure considered by WYlight. After computing $\delta(\alpha)$, an additional mining of $\mathcal{D}$ is performed to extract the significant patterns with p-value $\leq \delta(\alpha)$. As shown in (Llinares-López et al., 2015), WYlight significantly improves over FastWY in particular in terms of memory requirements, allowing the extraction of significant patterns from datasets larger than the ones that can be analyzed by FastWY.

3.2.3 Problem Definition

For a dataset $\mathcal{D}$ and the set of permutations $\mathcal{G}$, let $\delta(\alpha) = q_{\mathcal{H}}^{\mathsf{WY}}(\mathcal{G}, \mathcal{D}, \alpha)$ the threshold obtained through the WY permutation procedure when the bound on the $FWER$ is set to α. Let $p^{(k)}$ be the p-value of the k-th pattern with patterns sorted by (increasing) p-value. Given a dataset $\mathcal{D}$ and user-provided values k and α, our goal is to extract the set $TSP(\mathcal{D}, k, \alpha)$ of Top-k Statistically Significant Patterns with $FWER \leq \alpha$, defined as:

$$TSP(\mathcal{D}, k, \alpha) = \left\{\mathcal{P} : p_{\mathcal{P}}\left(\mathsf{z}_{\mathcal{D}^1}^{\mathcal{P}}\right) \leq \min\left\{\delta(\alpha), p^{(k)}\right\}\right\} \ .$$

Note that when less than k patterns have p-value below $\delta(\alpha)$, $TSP(\mathcal{D}, k, \alpha)$ contains all such patterns. In addition, according to our definition more than k patterns may be in $TSP(\mathcal{D}, k, \alpha)$, in case many have the same p-value $p^{(k)}$. We restrict our interest only to *closed* patterns, i.e. patterns whose children have support *strictly lower* than the one of the pattern itself, as non-closed patterns have the same p-values. Since the definition of closed pattern does not depend on the labels, restricting to closed patterns does not bias any analysis.

The following result establishes the required guarantees on false positives in $TSP(\mathcal{D}, k, \alpha)$ and it is a direct consequence of the fact that $TSP(\mathcal{D}, k, \alpha)$ is a subset of all the patterns that would be reported using the WY method.

Lemma 3.2.1. *The set $TSP(\mathcal{D}, k, \alpha)$ has $FWER \leq \alpha$.*

3.3 TopKWY Algorithm

In this section we present our algorithm TopKWY for mining the set $TSP(\mathcal{D}, k, \alpha)$. We first present its main strategy (Section 3.3.1) that can be applied to any pattern mining problem. We then analyze TopKWY showing theoretical evidence of the efficiency of its strategy (Section 3.3.2) and introduce improved bounds on the minimum attainable p-value used by TopKWY (Section 3.4). We also present extensions of TopKWY (Section 3.5) to control the *Generalized $FWER$*, to control the *False Discovery Proportion (FDP)*, and to employ different *exploration strategies* on the tree of candidate patterns. Finally, we introduce some crucial implementation details (Section 3.6), focusing on the problem of mining significant itemsets and subgraphs.

3.3.1 Main Strategy

TopKWY combines two key ideas. First, it maintains an estimate of $\delta_k = \min\{\delta(\alpha), p^{(k)}\}$ that is updated during the exploration of the patterns and maintains a corresponding minimum support threshold $\sigma = \psi^{-1}(\delta_k)$ that is raised during the exploration of the patterns. Analogously to the strategy employed by WYlight (Llinares-López et al., 2015) the updates of δ_k and σ depend on α, but in addition TopKWY

updates them also depending on the p-values of elements of $TSP(\mathcal{D}, k, \alpha)$. Second, the search tree of all possible patterns is explored in order of decreasing support, analogously to the strategy used by TopKMiner (Pietracaprina and Vandin, 2007) for mining top-k frequent patterns, which guarantees that only patterns of support greater or equal to *the final value of* σ (i.e., $\psi^{-1}(\delta_k)$) are explored.[1]

TopKWY is described in Algorithm 1. In line 1, the threshold δ_k is initialized to α (the threshold with no correction for multiple hypothesis) and σ is initialized accordingly to $\hat{\psi}^{-1}(\delta_k)$. All the elements of the set of minimum p-values $\{p_{\min}^{(j)}\}_{j=1}^{m}$ observed on the permuted datasets are initialized to 1 (their maximum achievable value) in line 2. The labels of the permuted datasets are generated in line 3. The pattern exploration is organized using a priority queue Q where each entry represents a pattern $\mathcal{P}$, with key equal to the support $\mathtt{z}_{\mathcal{D}}^{\mathcal{P}}$ and value representing all the information needed by the algorithm regarding $\mathcal{P}$ (e.g., $\mathtt{z}_{\mathcal{D}^1}^{\mathcal{P}}$) and also with relevant information regarding the parent $\mathtt{p}(\mathcal{P})$ of pattern $\mathcal{P}$ in the search tree (see Section 3.4 for more details). Q is initialized in line 4 and stores the frontier of unexplored patterns, keeping them accessible by non-increasing support. TopKWY stores patterns having p-value $\leq \delta_k$ in a priority queue $\mathcal{R}$, keeping them accessible by non-decreasing p-value. This is the set of candidates for $TSP(\mathcal{D}, k, \alpha)$, which are collected and produced in output as soon as possible during the exploration. This allows to reduce the memory requirements and to start analyzing the results during the exploration, without the need of waiting for the algorithm's termination. The first patterns in Q are obtained by the `expand(`$\mathcal{P}$`,`Q`)` operation on line 5 called on the empty pattern $\mathcal{P} = \emptyset$: this procedure generates all patterns children of the pattern $\mathcal{P}$ in the search tree (and their corresponding projected datasets), and inserts the ones of support $\geq \sigma$ in the queue Q. The details of efficient implementations of `expand` are described in Section 3.6. The `while` loop (lines 6-22) implements the main step of the exploration strategy: the most frequent pattern $\mathcal{P}$, its support $\mathtt{z}_{\mathcal{D}}^{\mathcal{P}}$, its support $\mathtt{z}_{\mathcal{D}^1}^{\mathcal{P}}$ in the minority label set $\mathcal{D}^1$ of $\mathcal{D}$, and the relevant information for its parent $\mathtt{p}(\mathcal{P})$ are extracted from Q in line 7. If the p-value $p_{\mathcal{P}} = p_{\mathcal{P}}\left(\mathtt{z}_{\mathcal{D}^1}^{\mathcal{P}}\right)$ of $\mathcal{P}$ is $p_{\mathcal{P}} \leq \delta_k$, then $\mathcal{P}$ is inserted in $\mathcal{R}$ as a potential result in line 11. In line 8, σ' is set to $\mathtt{z}_{\mathcal{D}}^{\mathcal{P}}$, which is an upper bound to the support of all elements stored in Q. This quantity is used to identify patterns surely in the set $TSP(\mathcal{D}, k, \alpha)$ without waiting for the final corrected significance threshold δ_k to be found, done in lines 12 and 13. k is updated accordingly, reducing it to the number of patterns which still need to be found. In order to compute the corrected significance threshold δ_k, the algorithm computes the p-values of pattern $\mathcal{P}$ in the m permuted datasets, updating the values of $\{p_{\min}^{(j)}\}_{j=1}^{m}$ if needed. This operation is done with the `test` procedure. Similarly to WYlight, our algorithm processes all the m permutations for every pattern $\mathcal{P}$ at once, computing only the needed exact

[1] This assumes that the search tree for patterns has the property that the children of a node have support not greater than the node itself, which is a usual property of pattern mining algorithms (Han et al., 2007; Uno et al., 2005; Nijssen and Kok, 2004) and is required by WYlight as well.

lower quantile of the set of minimum p-values of the WY permutations, and not the minimum p-values of every permuted dataset. Differently from WYlight, we use an improved lower bound $\psi'(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{p}(\mathcal{P}))$ to the minimum attainable p-value of $\mathcal{P}$ to decide (in line 14) whether to test $\mathcal{P}$ on the permuted datasets or not (see Section 3.4). This allows to skip the expensive counting of the supports of $\mathcal{P}$ on the set of samples $g^{(j)}(\mathcal{S})$ with permuted labels, for several patterns $\mathcal{P}$.

The significance threshold δ_k is decreased during the exploration in two cases: when the estimated $FWER$ for the current threshold δ_k increases above α (line 16), or when more than k patterns with p-value $\leq \delta_k$ are observed (line 18). The corresponding minimum support threshold σ is then updated accordingly in line 19. The correctness of these steps are proved in Section 3.3.2. After the update of δ_k and σ, elements which have become untestable are removed from Q in line 20, and elements which are not significant are removed from $\mathcal{R}$ in line 21. The current pattern $\mathcal{P}$ is expanded in line 22, and all its children having support $\geq \sigma$ are inserted into Q. The exploration ends when Q gets empty. When this happens, all elements still contained in $\mathcal{R}$ with p-value at most δ_k are reported as significant in line 23.

The strategy employed by TopKWY can be adapted to incrementally update k for the same α, providing an interactive mining process. This can be achieved by providing a maximum value $k^\star$ in input to Algorithm 1 to definitely prune untestable patterns, but freezing the computation after k patterns with p-value below the current value of $\hat{\psi}(\sigma)$ have been found. If the user wants to increase k, the exploration can continue without restarting the entire mining instance.

3.3.2 Analysis

Some important properties of TopKWY algorithm can be formally stated. The first regards the correctness of the algorithm.

Theorem 3.3.1 (Correctness of TopKWY)**.** *TopKWY outputs the set $TSP(\mathcal{D}, k, \alpha)$ of Top-k Significant Patterns with $FWER \leq \alpha$.*

Proof. The correctness of TopKWY follows from two observations: first, the final threshold δ_k obtained by the algorithm is correct; second, only patterns with p-value less or equal than the final value of δ_k are produced in output. We start by proving the first statement. δ_k is initialized to the value α, that is the uncorrected threshold for significance and is always $\geq \delta(\alpha)$. δ_k is decreased (and the corresponding minimum support threshold σ is increased) during the exploration in two cases. The first case (line 16) is when the estimated $FWER$ for the current threshold δ_k increases above α. This means that more than αm p-values $\{p_{\min}^{(j)}\}_{j=1}^{m}$ are below the current significance threshold $\delta_k = \hat{\psi}(\sigma)$, which allows for too many false positives, and the $FWER$ is not correctly controlled to the level α. δ_k is then updated to the highest value of δ for which the estimated $FWER$ is $\leq \alpha$. The second case is when more than k patterns with p-value $\leq \delta_k$ are observed (line 18). In this case, let $\tilde{p}$ be the highest p-value

Algorithm 1: TopKWY

Input: Transaction dataset $\mathcal{D}$ with class labels c, number of permutations m, target $FWER$ α, number of results k.

Output: Set of Top-k Significant Patterns with $FWER \leq \alpha$.

$\delta_k \leftarrow \alpha$; $\sigma \leftarrow \hat{\psi}^{-1}(\delta_k)$;
$p^{(j)}_{\min} \leftarrow 1$, $\forall j \in [1, m]$;
$\mathcal{G} \leftarrow$ sample m permutations from G;
$Q, \mathcal{R} \leftarrow$ empty priority queues;
`expand`($\emptyset$, Q);
while $Q \neq \emptyset$ **do**
  $(\mathcal{P}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}, \mathsf{p}(\mathcal{P})) \leftarrow Q$.`removeMax`();
  $\sigma' \leftarrow \mathsf{z}^{\mathcal{P}}_{\mathcal{D}}$;
  $p_{\mathcal{P}} \leftarrow p_{\mathcal{P}}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}\right)$;
  if $p_{\mathcal{P}} \leq \delta_k$ **then**
    $\mathcal{R}$.insert($\mathcal{P}, p_{\mathcal{P}}$);
  `/* ` $\mathcal{O} =$ `patterns surely in` $TSP(\mathcal{D}, k, \alpha)$ `*/`
  $\mathcal{O} \leftarrow \{\mathcal{P} \in \mathcal{R} : p_{\mathcal{P}} < \hat{\psi}(\sigma')\}$; produce $\mathcal{O}$ in output;
  $\mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{O}$; $k \leftarrow k - |\mathcal{O}|$;
  if $\psi'(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{p}(\mathcal{P})) \leq \delta_k$ **then**
    `test`($\mathcal{P}$, $\{p^{(j)}_{\min}\}_{j=1}^{m}$) ;
  `/* update` δ_k `based on estimate of` $\delta(\alpha)$ `*/`
  $\delta_k \leftarrow \min\{\delta_k, \max\{\delta : \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}\left[p^{(j)}_{\min} \leq \delta\right] \leq \alpha\}\}$;
  `/* update` δ_k `based on top-`k `patterns in` $\mathcal{R}$ `*/`
  $p^{(k)} \leftarrow k$-th largest p-value in $\mathcal{R}$;
  $\delta_k \leftarrow \min\{\delta_k, p^{(k)}\}$;
  `/* update` σ `*/`
  $\sigma \leftarrow \hat{\psi}^{-1}(\delta_k)$;
  `/* remove untestable patterns from` Q `*/`
  remove from Q all patterns $\mathcal{P}$ with $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}} < \sigma$;
  `/* remove non-significant patterns from` $\mathcal{R}$ `*/`
  remove from $\mathcal{R}$ all patterns $\mathcal{P}$ with $p_{\mathcal{P}} > \delta_k$;
  `expand`($\mathcal{P}$, Q);
produce in output $\{\mathcal{P} \in \mathcal{R} : p_{\mathcal{P}} \leq \delta_k\}$;

of the k most significant patterns observed up to this point. Then, all patterns of support $< \hat{\psi}^{-1}(\tilde{p})$ cannot result in a p-value $< \tilde{p}$ and therefore we need to consider (both in $\mathcal{D}$ and in the permuted datasets) only patterns of support at least $\hat{\psi}^{-1}(\tilde{p})$. That is, the minimum support threshold σ can be safely increased to $\hat{\psi}^{-1}(\tilde{p})$ with a corresponding significance threshold $\tilde{p}$. When δ_k is last updated, its value will then be equal to the minimum between $\delta(\alpha)$ and $p^{(k)}$.

We now prove the second statement. This is trivially correct for patterns produced in output by line 23. We then consider patterns produced in output in line 12. Note that the pattern $\mathcal{P}$ at a given iteration has support σ' and the search strategy employed by TopKWY guarantees that all patterns with support $> \sigma'$ have already been explored. Therefore, from this point on the algorithm will never encounter p-values $< \hat{\psi}(\sigma')$ and therefore the corrected significance threshold δ_k will be $\geq \hat{\psi}(\sigma')$. Thus all patterns in $\mathcal{R}$ with p-value $< \hat{\psi}(\sigma')$ can be safely produced in output (and removed from $\mathcal{R}$). □

The following result provides theoretical guarantees on which patterns will be explored by TopKWY, providing analytical evidence of the efficiency of our strategy.

Theorem 3.3.2 (Optimality of TopKWY)**.** *TopKWY expands only patterns of support* $\geq \hat{\psi}^{-1}(\delta_k)$.

Proof. Similarly to the proof of Thm. 3.3.1, when a pattern $\mathcal{P}$ of support σ' is extracted from Q, we are guaranteed that the algorithm will never encounter p-values $< \hat{\psi}(\sigma')$ again. Therefore the corrected significance threshold δ_k will be $\geq \hat{\psi}(\sigma')$, that is $\sigma' \geq \hat{\psi}^{-1}(\delta_k)$ (i.e., $\mathcal{P}$ is testable). □

We now show that in a simplified model for how p-values are obtained, there exists a family of datasets for which the expected difference between the number of patterns explored by a DFS strategy and the number of patterns explored by TopKWY is exponential in the size of the dataset. In the simplified model, the p-values obtained by random permutations are uniformly distributed in $(0, 1]$. (Note that we do not assume independence among p-values from different itemsets.) Consider now the family of datasets $\mathcal{D}_n = \{t_1, \ldots, t_n\}$ defined on the set of (binary) features $\mathcal{I} = \{i_1, \ldots, i_n\}$ where $t_j = \mathcal{I} \setminus \{i_j\}$. Moreover, half of transactions in $\mathcal{D}_n$ have label ℓ^0 while the other half have label ℓ^1.

Theorem 3.3.3. *Consider a dataset $\mathcal{D}_n$ from the family described above. Let ε be a constant such that $0 < \varepsilon \leq \frac{1}{2}$ and $\varepsilon n \in \mathbb{N}$. Assume that the choice of α and k is such $\delta(\alpha) = \hat{\psi}(\varepsilon n) = \binom{\frac{n}{2}}{\varepsilon n} / \binom{n}{\varepsilon n}$, and that m random permutations are used. Let X be the difference between the number of patterns explored by a DFS strategy and the number of patterns explored by TopKWY in the simplified model above. Then $\mathbb{E}[X] = \Omega\left(2^{\varepsilon n}/m\right)$.*

Proof. Note that in such dataset all patterns are closed. Let W be the set of patterns explored by TopKWY. The DFS strategy has to explore all the patterns in W (since they are testable). We now show that it will also explore, in expectation, $\Omega\left(2^{\varepsilon n}/m\right)$ additional patterns, that proves the statement.

It is easy to show that since $\varepsilon n \in \mathbb{N}$ and $\delta(\alpha) = \binom{\frac{n}{2}}{\varepsilon n} / \binom{n}{\varepsilon n}$, the set of testable patterns W is given by all patterns of size $\leq \varepsilon n$. For each pattern $\mathcal{P}_i$ and each j with $1 \leq j \leq m$, let X_{ij} be the random variable that is 1 if $\mathcal{P}_i$ has p-value $\leq \delta(\alpha)$ in the j-th permuted dataset, and 0 otherwise. Note that X_{ij} is a Bernoulli random variable of parameter $\delta(\alpha)$, for all i and j. Let Y be the number of p-values lower than $\delta(\alpha)$, assuming that the DFS has explored $\varepsilon n + N$ patterns. The expectation of Y is

$$\mathbb{E}\left[Y\right] = \mathbb{E}\left[\sum_{i=1}^{\varepsilon n+N}\sum_{j=1}^{m} X_{ij}\right] = \sum_{i=1}^{\varepsilon n+N}\sum_{j=1}^{m}\mathbb{E}\left[X_{ij}\right] = (\varepsilon n + N)m\delta(\alpha). \tag{3.2}$$

Note that requiring $\mathbb{E}\left[Y\right] \geq 1$ provides a lower bound to the number of p-values needed for the DFS to establish that $\delta(\alpha) = \binom{\frac{n}{2}}{\varepsilon n} / \binom{n}{\varepsilon n}$, since αm p-values below such threshold must be observed.

Let $v = \varepsilon n$. The condition $\mathbb{E}\left[Y\right] = (\varepsilon n + N)m\delta(\alpha) \geq 1$ implies

$$\begin{aligned}
(\varepsilon n + N) &\geq \frac{1}{m\delta(\alpha)} = \frac{1}{m}\frac{\binom{n}{v}}{\binom{\frac{n}{2}}{v}} = \frac{1}{m}\frac{n!(\frac{n}{2}-v)!}{(\frac{n}{2})!(n-v)!} \\
&= \frac{1}{m}\frac{(n-v)!\prod_{j=0}^{v-1}(n-j)}{(n-v)!}\frac{(\frac{n}{2}-v)!}{(\frac{n}{2}-v)!\prod_{j=0}^{v-1}(\frac{n}{2}-j)} \\
&= \frac{1}{m}\prod_{j=0}^{v-1}\frac{(n-j)}{(\frac{n}{2}-j)} \geq \frac{1}{m}\prod_{j=0}^{v-1} 2 = \frac{2^v}{m} = \frac{2^{\varepsilon n}}{m}
\end{aligned}$$

or, equivalently

$$N \geq \frac{2^{\varepsilon n}}{m} - \varepsilon n \in \Omega\left(2^{\varepsilon n}/m\right).$$

Note that since the set of testable patterns includes all patterns of size $\leq \varepsilon n$, all the $\Omega\left(2^{\varepsilon n}/m\right)$ patterns explored by the DFS *after* exploring the first εn patterns and before observing the first p-value below $\delta(\alpha)$ have size $> \varepsilon n$ and, thus, are not in W, which proves the statement. □

Compared to a depth first search (DFS) exploration strategy (i.e., the one employed by WYlight), the *best first* exploration strategy followed by TopKWY has the additional costs required by operations involving data structures $\mathcal{R}$ and Q. $\mathcal{R}$ can be implemented as a heap of entries (p, ℓ_p), where the key p is a p-value and the value ℓ_p is a list of patterns which contains all the patterns with the same p-value p. With such implementation, operations involving $\mathcal{R}$ (lines 11, 12, 17, and 21) can be perfomed with $\mathcal{O}\left(\log k\right)$ operations, since $\mathcal{R}$ will contain at most k entries. Analogously,

Q can be implemented as a heap of entries (x, ℓ_x), where the key x is a support and the value ℓ_x is a list of patterns with the same value of $\hat{\psi}(x)$ (that is, having the same support x). With such implementation, operations involving Q (lines 7, 20, and 22) can be performed with $\mathcal{O}(\log n)$ operations (n is the number of transactions in $\mathcal{D}$). Alternatively, $\mathcal{R}$ and Q can be implemented so that all operations require time $\mathcal{O}(1)$ by storing references to lists ℓ_p and ℓ_x in arrays of size $\mathcal{O}(n^2)$ (since the p-value of $\mathcal{P}$ is a function of the support $\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}$ of $\mathcal{P}$ and the number $\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}$ of transactions with label ℓ^1 and containing $\mathcal{P}$) and $\mathcal{O}(n)$, respectively. (Note that this additional space requirement is not always impractical, since $\mathcal{O}(nm)$ space is needed to store the permuted labels.) We also note that computing the improved lower bound $\psi'(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{p}(\mathcal{P}))$ (line 14), has the same cost as computing the lower bound $\hat{\psi}(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}})$ to the p-value used in WYlight (see Section 3.4). Even with the additional costs required by $\mathcal{R}$ and Q, the best first strategy of TopKWY leads to significant improvements in running time, as demonstrate by our experimental evaluation (Section 3.7).

3.4 Improved Bounds on Minimum Attainable p-value

In this section we prove novel and efficiently computable lower bounds on the minimum p-value achievable by a pattern $\mathcal{P}$ that are tighter than the ones introduced by LAMP (Terada et al., 2013a) and are of particular interest in the context of WY permutation testing. These bounds are based on information computed when processing a *parent pattern* Y of $\mathcal{P}$; in the case of itemsets, Y is a parent of $\mathcal{P}$ (or, alternatively, $\mathcal{P}$ is a *child* of Y) when $Y \subset \mathcal{P}$. Such bounds can be used to *skip* the expensive processing of the permutations for $\mathcal{P}$ when they ensure that it is not possible to improve the current estimate of the corrected significance threshold. While we present these bounds as a critical component of TopKWY, they may be of independent interest since can be employed in WYlight or similar algorithms to speed-up WY permutation testing.

Let the pattern $\mathcal{P}$ be a child of Y, that is $\mathcal{P} \supset Y$. Then $\mathsf{z}_{\mathcal{D}}^{Y} \geq \mathsf{z}_{\mathcal{D}^1}^{Y} \geq 0$ and $\mathsf{z}_{\mathcal{D}}^{Y} \geq \mathsf{z}_{\mathcal{D}}^{\mathcal{P}}$. Since the set of transactions (i.e., the *conditional dataset*) containing $\mathcal{P}$ is a subset of the set of transactions containing Y, we can bound the support $\mathsf{z}_{\mathcal{D}^1}^{\mathcal{P}}$ of $\mathcal{P}$ in the set of samples $\mathcal{D}^1$ with the following relations:

$$\max(\mathsf{z}_{\mathcal{D}^1}^{Y} - (\mathsf{z}_{\mathcal{D}}^{Y} - \mathsf{z}_{\mathcal{D}}^{\mathcal{P}}), 0) \leq \mathsf{z}_{\mathcal{D}^1}^{\mathcal{P}} \leq \min(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_{\mathcal{D}^1}^{Y}).$$

Considering the m permuted class labels, let z_j^Y be the number of transactions containing Y with label ℓ^1 (i.e., z_j^Y is the value of $\mathsf{z}_{\mathcal{D}^1}^{Y}$ when the class labels are given by the j-th permutation). An analogous relation holds between $\mathsf{z}_j^{\mathcal{P}}$ and z_j^Y, for all j:

$$\max(\mathsf{z}_j^{Y} - (\mathsf{z}_{\mathcal{D}}^{Y} - \mathsf{z}_{\mathcal{D}}^{\mathcal{P}}), 0) \leq \mathsf{z}_j^{\mathcal{P}} \leq \min(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_j^{Y}).$$

An immediate consequence of these bounds on $\mathsf{z}_j^{\mathcal{P}}$ are *lower bounds* to $p_{\mathcal{P}}\left(\mathsf{z}_j^{\mathcal{P}}\right)$.

Lemma 3.4.1. *Let* $\check{\mathsf{z}}_j^{\mathcal{P}} = \max(\mathsf{z}_j^Y - (\mathsf{z}_{\mathcal{D}}^Y - \mathsf{z}_{\mathcal{D}}^{\mathcal{P}}), 0)$ *and* $\check{\mathsf{z}}_j^{\mathcal{P}} = \min(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_j^Y)$. *Then, for all* $j \in [1, m]$,

$$p_{\mathcal{P}}\left(\mathsf{z}_j^{\mathcal{P}}\right) \geq \min\left\{p_{\mathcal{P}}\left(\check{\mathsf{z}}_j^{\mathcal{P}}\right), p_{\mathcal{P}}\left(\hat{\mathsf{z}}_j^{\mathcal{P}}\right)\right\} \ .$$

This result suggests that if we have already computed $\mathsf{z}_j^Y, \forall j \in [1, m]$ while processing the permuted labels of the conditional dataset of Y, we could skip the expensive computation of $\mathsf{z}_j^{\mathcal{P}}$, and, therefore, $p_{\mathcal{P}}\left(\mathsf{z}_j^{\mathcal{P}}\right), \forall j \in [1, m]$, in situations when the lower bounds to $p_{\mathcal{P}}\left(\mathsf{z}_j^{\mathcal{P}}\right)$ are *greater* than the current value of the corrected significance threshold. In the following, we present a bound valid for all $p_{\mathcal{P}}\left(\mathsf{z}_j^{\mathcal{P}}\right)$ simultaneously, that is a function of only the minimum and maximum elements of z_j^Y, instead of all of them. Let

$$\mathsf{z}_{\min}^Y = \min\left\{\mathsf{z}_j^Y : j \in [1, m]\right\}$$

and

$$\mathsf{z}_{\max}^Y = \max\left\{\mathsf{z}_j^Y : j \in [1, m]\right\} \ .$$

Then, $\forall j \in [1, m]$ we bound $\mathsf{z}_j^{\mathcal{P}}$ as:

$$\mathsf{z}_{\min}^{\mathcal{P}} = \max(\mathsf{z}_{\min}^Y - (\mathsf{z}_{\mathcal{D}}^Y - \mathsf{z}_{\mathcal{D}}^{\mathcal{P}}), 0) \leq \mathsf{z}_j^{\mathcal{P}} \leq \min(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_{\max}^Y) = \mathsf{z}_{\max}^{\mathcal{P}} \ .$$

This allows to compute a bound $\psi'(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_{\mathcal{D}}^Y, \mathsf{z}_{\min}^Y, \mathsf{z}_{\max}^Y)$ to the minimum attainable p-value of $\mathcal{P}$ that is tighter than $\psi(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}})$:

$$\psi'(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_{\mathcal{D}}^Y, \mathsf{z}_{\min}^Y, \mathsf{z}_{\max}^Y) = \min\left\{p_{\mathcal{P}}\left(\mathsf{z}_{\min}^{\mathcal{P}}\right), p_{\mathcal{P}}\left(\mathsf{z}_{\max}^{\mathcal{P}}\right)\right\} \ . \tag{3.3}$$

The bound in Equation 3.3 is evaluated in constant time, assuming $p_{\mathcal{P}}(x)$ is precomputed for all valid values of x, as done in WYlight and in TopKWY to efficiently implement the `test` function. The following are simple consequences of Lemma 3.4.1 and the fact that $\mathsf{z}_{\min}^{\mathcal{P}}$ and $\mathsf{z}_{\max}^{\mathcal{P}}$ are always equally or more tight than the naive bounds on $\mathsf{z}_{\mathcal{D}^1}^{\mathcal{P}}$ assumed by $\psi(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}})$.

Lemma 3.4.2. $\min\left\{p_{\mathcal{P}}\left(\mathsf{z}_j^{\mathcal{P}}\right) : j \in [1, m]\right\} \geq \psi'(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_{\mathcal{D}}^Y, \mathsf{z}_{\min}^Y, \mathsf{z}_{\max}^Y) \geq \psi(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}})$.

If for the current value of the significance threshold δ_k it holds that $\psi'(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_{\mathcal{D}}^Y, \mathsf{z}_{\min}^Y, \mathsf{z}_{\max}^Y) > \delta_k$, then we can infer, without computing $\{\mathsf{z}_j^{\mathcal{P}}\}_{j=1}^m$, that none of the m p-values of $\mathcal{P}$ in the permuted datasets will improve the estimate of the current lower-quantile of the set $\{p_{\min}^{(j)}\}_{j=1}^m$ and therefore cannot contribute to the computation of $\delta(\alpha)$ or δ_k. That is, all the computation on the permuted datasets can be skipped for the current pattern $\mathcal{P}$. For all children of $\mathcal{P}$, if $\mathcal{P}$ is not tested the bounds $\mathsf{z}_{\min}^{\mathcal{P}}$ and $\mathsf{z}_{\max}^{\mathcal{P}}$ can be propagated to compute bounds also on their class distribution; if $\mathcal{P}$ is tested, then we propagate the actual minimum and maximum values of $\{\mathsf{z}_j^{\mathcal{P}}\}_{j=1}^m$. In Algorithm 1 we use the bound above with the values propagated by

the parent $\mathsf{p}(\mathcal{P})$ of $\mathcal{P}$ and use $\psi'(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{p}(\mathcal{P}))$ to highlight this fact. This optimization is particularly effective when patterns have a high degree of correlation, i.e., when patterns share many transactions.

Note that even if $\mathcal{P}$ does not need to be tested, descendants of $\mathcal{P}$ may need to be tested. However, using the bound $\psi'(\cdot)$ we can quickly identify cases in which none of the descendants of $\mathcal{P}$ need to be explored and therefore the entire subtree can be pruned. In particular, since all the descendants of $\mathcal{P}$ will have support $\leq \mathsf{z}_{\mathcal{D}}^{\mathcal{P}} - 1$, considering $\mathsf{z}_{\mathcal{D}^1}^{\mathcal{P}}$ (i.e., the number of transactions containing $\mathcal{P}$ and with label ℓ^1 in the dataset $\mathcal{D}$), the algorithm can find $\min\{\psi'(i, \mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_{\min}^{\mathcal{P}}, \mathsf{z}_{\max}^{\mathcal{P}}) : i \in [\sigma, \mathsf{z}_{\mathcal{D}}^{\mathcal{P}} - 1]\}$, and if such value is $> \delta_k$ we can prune all the search subtrees rooted in the children of $\mathcal{P}$. This optimization is part of the `expand` operation in TopKWY. These novel bounds consider the information of one common ancestor pattern to avoid useless computations for many of its children: in practice, the number of tests to perform across the permuted datasets can be significantly smaller than the number of testable patterns, leading to a significant computational speed-up. The approach above can extended to bound $\min\left\{p_{\mathcal{P}}\left(\mathsf{z}_j^{\mathcal{P}}\right) : j \in [1, m]\right\}$ by considering the information computed on the intersection of the conditional datasets of any pair of patterns $\mathcal{P}$ and Y, even if $\mathcal{P} \not\supset Y$.

Another possible extension of the techniques we derive in this section is to consider not only *one pair* $(\mathsf{z}_{\min}^{Y}, \mathsf{z}_{\max}^{Y})$, but v *pairs* $(\mathsf{z}_{\min}^{Y,i}, \mathsf{z}_{\max}^{Y,i})$, for all $i \in [1, v]$: for each i, we define the set J_i as a subset of $\{1, \ldots, m\}$, with $\bigcup_{i=1}^{v} J_i = \{1, \ldots, m\}$. For all i, we bound all values z_j^{Y} for all $j \in J_i$ with $\mathsf{z}_{\min}^{Y,i}$ and $\mathsf{z}_{\max}^{Y,i}$. If we define

$$\mathsf{z}_{\min}^{Y,i} = \min\left\{\mathsf{z}_j^{Y} : j \in J_i\right\} \quad , \quad \mathsf{z}_{\max}^{Y,i} = \max\left\{\mathsf{z}_j^{Y} : j \in J_i\right\},$$

$$\mathsf{z}_{\min}^{\mathcal{P},i} = \max\left\{\mathsf{z}_{\max}^{Y,i} - (\mathsf{z}_{\mathcal{D}}^{Y} - \mathsf{z}_{\mathcal{D}}^{\mathcal{P}}), 0\right\} \quad , \quad \mathsf{z}_{\max}^{\mathcal{P},i} = \min\left\{\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_{\max}^{Y,i}\right\},$$

then we simply obtain, for all i,

$$\min\left\{p_{\mathcal{P}}\left(\mathsf{z}_j^{\mathcal{P}}\right) : j \in J_i\right\} \geq \min\left\{p_{\mathcal{P}}\left(\mathsf{z}_{\min}^{\mathcal{P},i}\right), p_{\mathcal{P}}\left(\mathsf{z}_{\max}^{\mathcal{P},i}\right)\right\}.$$

This provides a trade-off between the memory (required to store the $2v$ bounds on the expanded nodes of the search space) and the time to evaluate the bound (that is linear in the number v of sets instead of constant), and the time the algorithm saves by skipping computations of the permutations, that is a direct consequence of how tight the bounds on the p-values are; in fact, the permutations that have to be processed are only the ones belonging to the sets J_i such that $\min\left\{p_{\mathcal{P}}\left(\mathsf{z}_{\min}^{\mathcal{P},i}\right), p_{\mathcal{P}}\left(\mathsf{z}_{\max}^{\mathcal{P},i}\right)\right)$ is not higher than the current value of the significance threshold.

3.5 Extensions of TopKWY

3.5.1 Controlling the Generalized $FWER$

While the main focus of TopKWY is to control the $FWER$ of the output, a simple modification provides an algorithm to control the *generalized* $FWER$ (g-$FWER$ (Lehmann and Romano, 2012)). The g-$FWER$ is defined as the probability that at least g false positives are reported in output. In several applications one may be willing to tolerate a small amount of false discoveries in order to increase the power of detecting significant patterns, provided the number of false discoveries can be controlled. In such cases methods to discover significant patterns while controlling the g-$FWER$ are preferred to methods controlling $FWER$. Let FP be the number of false positives reported by a certain procedure $\mathcal{A}$ using $\mathcal{D}$, then,

$$g\text{-}FWER = \Pr\left(FP \geq g\right) \quad .$$

Let g-$FWER(\delta)$ be the g-$FWER$ obtained using δ as *corrected* significance threshold, that is by flagging as significant patterns with p-value $\leq \delta$. Note that the Westfall-Young procedure can be used to estimate g-$FWER(\delta)$ as

$$g\text{-}FWER(\delta) = \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}[p_g^{(j)} \leq \delta]$$

where $p_g^{(j)}$ is the g-th smallest p-value (over all patterns) in the j-th permuted dataset.

Algorithm 1 can be simply modified to obtain the set of Top-k Significant Patterns with g-$FWER \leq \alpha$. To achieve this, it is sufficient to perform the following changes: replace $\texttt{test}(\mathcal{P}, \{p_{\min}^{(j)}\}_{j=1}^{m})$ (line 15) with $\texttt{test}(\mathcal{P}, \{p_g^{(j)}\}_{j=1}^{m})$, where $\texttt{test}(\mathcal{P}, \{p_g^{(j)}\}_{j=1}^{m})$ computes the p-values of pattern $\mathcal{P}$ in the m permuted datasets and updates the values of $\{p_g^{(j)}\}_{j=1}^{m}$ if needed; replace line 16 with "$\delta_k \leftarrow \max\{\delta : g\text{-}FWER(\delta) \leq \alpha\}$". Let TopKWY-$g$ be such modified algorithm. We have the following.

Lemma 3.5.1. *TopKWY-g outputs the set $TSP(\mathcal{D}, k, \alpha)$ of Top-k Significant Patterns with g-$FWER \leq \alpha$.*

The proof is analogous to the proof of Theorem 3.3.1.

In addition to finding the top-k most significant pattern with bounded g-$FWER$, with g provided in input, TopKWY-g can be adapted to a different scenario. In this case, one may want to retrieve the k most significant patterns, using the random permutations to obtain a rigorous *estimate* of how many of such results are likely to be false positives. More formally, for k and α provided by the user, one may be interested in computing the quantity $g^\star$ defined as

$$g^\star = \min\left\{g : g\text{-}FWER(p^k) \leq \alpha\right\},$$

that is the *mininum* value of g such that the g-$FWER$ is controlled when the significance threshold is $p^{(k)}$, that is the highest p-value of the set of top-k results we extracted. $g^\star$ provides useful knowledge on the *quality* of the set of top-k results provided to the user. In this situation the user is not required to fix a-priori g before examining the data. Further simple modifications to TopKWY-g are sufficient to obtain such variant, that are: remove line 16 and replace line 23 with "produce in output $\{\mathcal{P} \in \mathcal{R} : p_\mathcal{P} \leq \delta_k\}$ and $g^\star = \min\left\{g : g\text{-}FWER(p^k) \leq \alpha\right\}$". Let TopKWY* be such modified algorithm; we obtain the following guarantees.

Lemma 3.5.2. *TopKWY* outputs the set of patterns $\left\{\mathcal{P} : p_\mathcal{P}\left(\mathsf{z}^\mathcal{P}_{\mathcal{D}^1}\right) \leq p^k\right\}$ of Top-k Significant Patterns and $g^\star = \min\left\{g : g\text{-}FWER(p^k) \leq \alpha\right\}$.*

3.5.2 Bounding the Proportion of False Discoveries

The False Discovery Proportion (FDP) (van der Laan et al., 2004; Lehmann and Romano, 2012) of a set of rejected hypotheses $\mathcal{V}$ is defined as the ratio $FD/|\mathcal{V}|$, where FD is the (unknown) number of false discoveries $\in \mathcal{V}$; note that when $|\mathcal{V}| = 0$, the FDP is 0. Let $\zeta, \alpha \in (0, 1)$ and $k, g \in [1, +\infty)$. Define the set $\mathcal{V} = \{(\mathcal{P}, p_\mathcal{P})\}$ containing pairs where $\mathcal{P}$ is a pattern and $p_\mathcal{P}$ its p-value in $\mathcal{D}$, such that all the following conditions hold:

$$\begin{aligned}
&\max\{p_\mathcal{P} : (\mathcal{P}, p_\mathcal{P}) \in \mathcal{V}\} \leq \delta^\star,\\
&\delta^\star = \max\{\delta : g\text{-}FWER(\delta) \leq \alpha\},\\
&g \leq \zeta|\mathcal{V}|, \quad |\mathcal{V}| \leq k.
\end{aligned}$$

It is possible to prove that a set $\mathcal{V}$ satisfying the above conditions has size at most k and $FDP \leq \zeta$ with probability $\geq 1 - \alpha$. Simple modifications to TopKWY-g lead to an algorithm that outputs $\mathcal{V}$ with the aforementioned guarantees: remove lines 12 and 13, replace line 16 with "$\delta_k \leftarrow \max\{\delta : (\lfloor k\zeta \rfloor)\text{-}FWER(\delta) \leq \alpha\}$" and line 23 with "produce in output $\mathcal{V}(\delta^\star) = \{\mathcal{P} \in \mathcal{R} : p_\mathcal{P} \leq \delta^\star\}$ where $\delta^\star = \max\{\delta : (\lfloor \zeta|\mathcal{V}(\delta)| \rfloor)\text{-}FWER(\delta) \leq \alpha\}$". Let such algorithm be TopKWY-ζ. We obtain the following result.

Lemma 3.5.3. *TopKWY-ζ outputs the set of patterns $\mathcal{V}$ of size at most k with False Discovery Proportion $\leq \zeta$ with probability $\geq 1 - \alpha$.*

3.5.3 Alternative Exploration Strategies

While TopKWY builds on examining the search tree of all possible patterns in order of decreasing support, i.e. with a *best first strategy* analogous to the one used by TopKMiner (Pietracaprina and Vandin, 2007) for mining top-k frequent patterns, it can also be modified to efficiently obtain the set $TSP(\mathcal{D}, k, \alpha)$ using different exploration strategies, e.g. a level-wise exploration of the search tree (performed, e.g.,

by the Apriori algorithm (Agrawal and Srikant, 1994) for itemsets) or a depth first search on the tree of all possible patterns (Uno et al., 2005; Nijssen and Kok, 2004). This can be achieved by setting σ' to $\max\left\{\mathsf{z}_{\mathcal{D}}^{\mathcal{P}} : \mathcal{P} \in Q\right\}$ (instead that to $\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}$) in line 8 of Algorithm 1 and by an appropriate choice of the priority for patterns in the priority queue Q, that stores the frontier of unexplored patterns: to obtain a level-wise exploration for itemsets, the priority of pattern $\mathcal{P}$ is set to the total number $|\mathcal{I}|$ of items minus $|\mathcal{P}|$; to obtain a depth first search, the priority of patterns $\mathcal{P}$ is set to its level in the search tree.

While for strategies other then the *best first* one the optimality (Theorem 3.3.2) is not guaranteed, the possibility to employ other strategies allows to obtain the Top-k Significant Patterns starting from efficient implementations of frequent pattern mining algorithms that build on such strategies for various types of patterns (e.g., subgraphs (Nijssen and Kok, 2004)).

3.6 Implementation Details

An efficient implementation of *expand* and *test* procedures is critical for the efficiency of TopKWY. This crucially depends on the representation of $\mathcal{D}$ and the permuted class labels, and both depend on the type of patterns of interest. In Sections 3.6.1 and 3.6.2 we now describe in more details the implementations for significant itemsets mining and for significant subgraphs mining; in particular, for significant itemsets we discuss the implementation of TopKWY as described in Section 3.3.1 as well as the variant, described in Section 3.5.3 of TopKWY based on the DFS strategy, which we denote by TopKWY-dfs; for significant subgraphs we only consider the implementation of TopKWY-dfs.

3.6.1 Significant Itemset Mining

Our implementation of TopKWY is based upon TopKMiner (Pietracaprina and Vandin, 2007), which mines top-k frequent closed itemsets. As for TopKMiner, TopKWY uses a PatriciaTrie (Zandolin and Pietracaprina, 2003) to store a compact representation of the dataset $\mathcal{D}$ in which transactions sharing the same prefix are represented by the same node in the tree. The conditional dataset of (i.e., the set transactions containing) an itemset Y is stored as a list d_Y of nodes of the PatriciaTrie. An additional counter is added to every node, representing how many transactions with prefix represented by the node have label ℓ^1. The same is done for the m permutations adding m counters to each node in the trie. Since the PatriciaTrie is built adding one transaction at a time, a technique similar to reservoir sampling is used to generate the m permuted labels of every transaction. Let $\mathbf{r}$ be a vector of length m, with all components $\mathbf{r}^{(j)}, j = 1, \ldots, m$ of $\mathbf{r}$ initialized to n_1, the number of transactions to assign labels ℓ^1 for every permutation of index j. For every transaction t_i, with $i \in [1, n]$, the j-th label of t_i is set as ℓ^1 with probability $\mathbf{r}^{(j)}/(n - i + 1)$, ℓ^0

otherwise. If ℓ^1 is chosen, then $\mathbf{r}^{(j)}$ is decreased by one. This method guarantees that the total number of transactions with label ℓ^1 will be n_1 for every $j \in [1, m]$ and that the labels of the j-th permuted dataset are obtained by a uniformly chosen random permutation of the class labels.

Our implementation of TopKWY-dfs is based upon LCM 3 (Uno et al., 2005), which mines frequent closed itemsets using a depth first strategy. The third version of LCM combines various techniques and data structures to accelerate the generation and the computation of the frequencies of frequent closed itemsets.

3.6.2 Significant Subgraph Mining

Our implementation of TopKWY-dfs relies on Gaston (Nijssen and Kok, 2004) to mine significant subgraphs. Gaston first considers simple patterns, such as paths and trees, since efficient techniques for isomorphism checking are available for such acyclic structures. Only after this first phase, denoted as "quickstart", general subgraphs, containing cycles, are evaluated. The search strategy of Gaston relies on a depth first enumeration of subgraphs. We do not provide a subgraph variant of TopKWY because no competitive algorithms based on a best first exploration strategy are currently available (Wörlein et al., 2005; Nijssen and Kok, 2006).

3.7 Experimental Evaluation

We implemented and tested TopKWY and TopKWY-dfs for the extraction of significant itemsets and significant subgraphs. Our experimental evaluation has three goals. First, to assess the number of significant patterns found in real datasets. Second, to evaluate the performance of TopKWY: since no other tool for the extraction of Top-k Significant Patterns exists, we compare TopKWY and TopKWY-dfs with the state-of-the-art tool for significant pattern mining, WYlight (Llinares-López et al., 2015). While the techniques introduced in this work can be extended to other multiple hypothesis testing procedures, such as LAMP (Terada et al., 2013a), we do not compare with LAMP or derived strategies (Minato et al., 2014) since Llinares-López et al. (2015) have shown that WY permutation testing results in higher power. Third, to assess the impact of our improved bounds and implementation choices on performances.

In Section 3.7.1 we describe the implementation and computational environment for our experiments. In Section 3.7.2 we describe the datasets we used. In Section 3.7.3 we describe the experiments we have performed and our choice of parameters. Finally, in Section 3.7.4 we report and discuss the results of our experiments.

Table 3.1: Itemset Datasets statistics. For each dataset the table reports: the number $|\mathcal{D}|$ of transactions; the number $|\mathcal{I}|$ of items; the average transaction length avg; the fraction n_1/n of transactions with label ℓ^1; the number $SP(0.05)$ of significant patterns for $FWER = 0.05$.

dataset	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{I}\rvert$	avg	n_1/n	$SP(0.05)$
svmguide3(L)	1,243	44	21.9	0.23	36,736
chess(U)	3,196	75	37	0.05	$> 10^7$
mushroom(L)	8,124	118	22	0.48	71,945
phishing(L)	11,055	813	43	0.44	$> 10^7$
breast cancer(L)	12,773	1,129	6.7	0.09	6
a9a(L)	32,561	247	13.9	0.24	348,611
pumb-star(U)	49,046	7117	50.5	0.44	$> 10^7$
bms-web1(U)	58,136	60,978	2.51	0.03	704,685
connect(U)	67,557	129	43	0.49	$> 10^8$
bms-web2(U)	77,158	330,285	4.59	0.04	289,012
retail(U)	88,162	16,470	10.3	0.47	3,071
ijcnn1(L)	91,701	44	13	0.10	607,373
T10I4D100K(U)	100,000	870	10.1	0.08	3,819
T40I10D100K(U)	100,000	942	39.6	0.28	5,986,439
codrna(L)	271,617	16	8	0.33	4,088
accidents(U)	340,183	467	33.8	0.49	$> 10^7$
bms-pos(U)	515,597	1,656	6.5	0.40	26,366,131
covtype(L)	581,012	64	11.9	0.49	542,365
susy(U)	5,000,000	190	43	0.48	$> 10^7$

3.7.1 Implementation and Environment

We implemented TopKWY in C++ as an extension of the TopKMiner algorithm (Pietracaprina and Vandin, 2007). For TopKWY-dfs we modified the C implementation of WYlight (based on LCM (Uno et al., 2005) and Gaston (Nijssen and Kok, 2004))[2]. All implementations were compiled with `gcc 4.8.4`. Our experiments have been performed on a 2.30 GHz Intel Xeon CPU machine with 512 GB of RAM, running on Ubuntu 14.04. Our code and scripts to replicate all experiments described in the paper are available at `https://github.com/VandinLab/TopKWY`.

[2]Available at `https://github.com/fllinares/wylight`

Table 3.2: Subgraph Datasets statistics. For each dataset the table reports: the number $|\mathcal{D}|$ of graphs; the average number of nodes $|V_{avg}|$; the average number of edges $|E_{avg}|$; the fraction n_1/n of graphs with label ℓ^1; the number $SP(0.05)$ of significant patterns for $FWER = 0.05$. The number in brackets report the maximum number of vertexes of the explored subgraphs, that has been limited to allow practical running times for the experiments.

dataset	$\|\mathcal{D}\|$	V_{avg}	E_{avg}	n_1/n	$SP(0.05)$
MUTAG	188	17.93	19.79	19.79	70,184
BZR(30)	405	35.75	38.36	0.21	80,425
COX2	467	41.22	43.45	0.22	$> 10^6$
ENZYMES(10)	600	32.63	62.14	0.17	112,158
DHFR(30)	753	42.43	44.54	0.61	$> 10^6$
DD	1,178	284.32	715.66	0.58	80,256
AIDS(30)	2,000	15.69	16.20	0.2	566,727
NCI1	4,110	29.87	32.30	0.5	$> 10^6$
NCI109	4,127	29.68	32.13	0.5	$> 10^6$
Mutagenicity	4,337	30.32	30.77	0.44	$> 10^6$
Tox_21_AHR(30)	8,169	18.09	18.50	0.12	98,398

3.7.2 Datasets

For itemsets mining, we performed our experiments using 19 datasets: the 10 largest ones used in (Llinares-López et al., 2015) and available at FIMI'04[3] and UCI[4], all the datasets used in (Komiyama et al., 2017), available from the libSVM repository[5], and 4 additional ones (a9a, bms-web1, accidents, susy) available from libSVM, FIMI'04, and SPMF[6]. The datasets' statistics are in Table 3.1. For each dataset, we also note if it already contained labels (L) or not (U). For unlabeled datasets we simulated a typical analysis requiring to find itemsets correlated with a given item (feature) in a dataset. For every unlabeled dataset 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 define the target label. The reported ratio n_1/n for the minority class of unlabeled datasets refers to the output of this labeling process. For real-valued features we obtained two bins by thresholding at the mean value and using one item for each bin (analogously to Komiyama et al. (2017)).

For subgraphs mining, we considered 11 of the largest datasets with binary target

[3] `http://fimi.ua.ac.be`

[4] `https://archive.ics.uci.edu/ml/index.php`

[5] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

[6] `http://www.philippe-fournier-viger.com/spmf`

labels available from a repository[7] of benchmark datasets; most of them are also analysed by Llinares-López et al. (2015). The datasets' statistics are in Table 3.2. In some cases, we bounded the maximum number of vertexes of the subgraphs that are explored by Gaston, in order to obtain practical running times for the experiments. Such limits are reported in the parenthesis after the dataset's name in Table 3.2.

3.7.3 Parameters and Experiments

For TopKWY and TopKWY-dfs we considered $k = 10^i$ for $i \in [1, 6]$. For all the datasets we analyzed, we ran TopKWY and TopKWY-dfs, for all such values of k, and WYlight. We fixed the number of permutations $m = 10^4$, shown to be a good choice by Llinares-López et al. (2015), and fixed the commonly used value $\alpha = 0.05$ as $FWER$ threshold. For the comparison between TopKWY, TopKWY-dfs, and WYlight, we repeated every experiment 10 times, recording the running time and peak memory provided by the operating system; we report the averages over the 10 runs, standard deviations are negligible and therefore not shown. The measures reported for TopKWY include the time and space to retrieve statistically significant patterns and write them on file, while for TopKWY-dfs and WYlight we only report the time and space needed to find the optimal significance threshold, which corresponds to the first step of the method, therefore reporting a lower bound to their runtimes. We stopped the execution of an algorithm if it did not conclude after (at least) one month of computation; for these cases, the indicated time and peak memory are lower bounds. For experiments testing the impact of parameters or implementation choices on TopKWY we used only one execution.

3.7.4 Results

Itemsets Mining

Table 3.1 reports the number of significant patterns for $\alpha = 0.05$ in the datasets we considered, obtained by running TopKWY (with $k = +\infty$) or WYlight. For some datasets we stopped the computation after 1 month, so only a lower bound is available. In most cases, the number of significant patterns is extremely large: for 11 out of 19 datasets there are $> 5 \times 10^5$ significant patterns and in 7 datasets there are $> 10^7$ significant patterns. Therefore a direct way to limit the number of significant patterns in output, as provided by the Top-k Significant Patterns, is required.

Figure 3-1 compares the running time of TopKWY and WYlight. Note that for the 11 datasets in which the number of significant patterns is $< 10^6$, TopKWY with $k = 10^6$ identifies all the significant patterns and produces the same patterns found with WYlight. For 15 out of 19 datasets, TopKWY (with $k = 10^6$) is faster than WYlight by a factor at least 2. For 9 datasets TopKWY is faster than WYlight by

[7] `https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets`

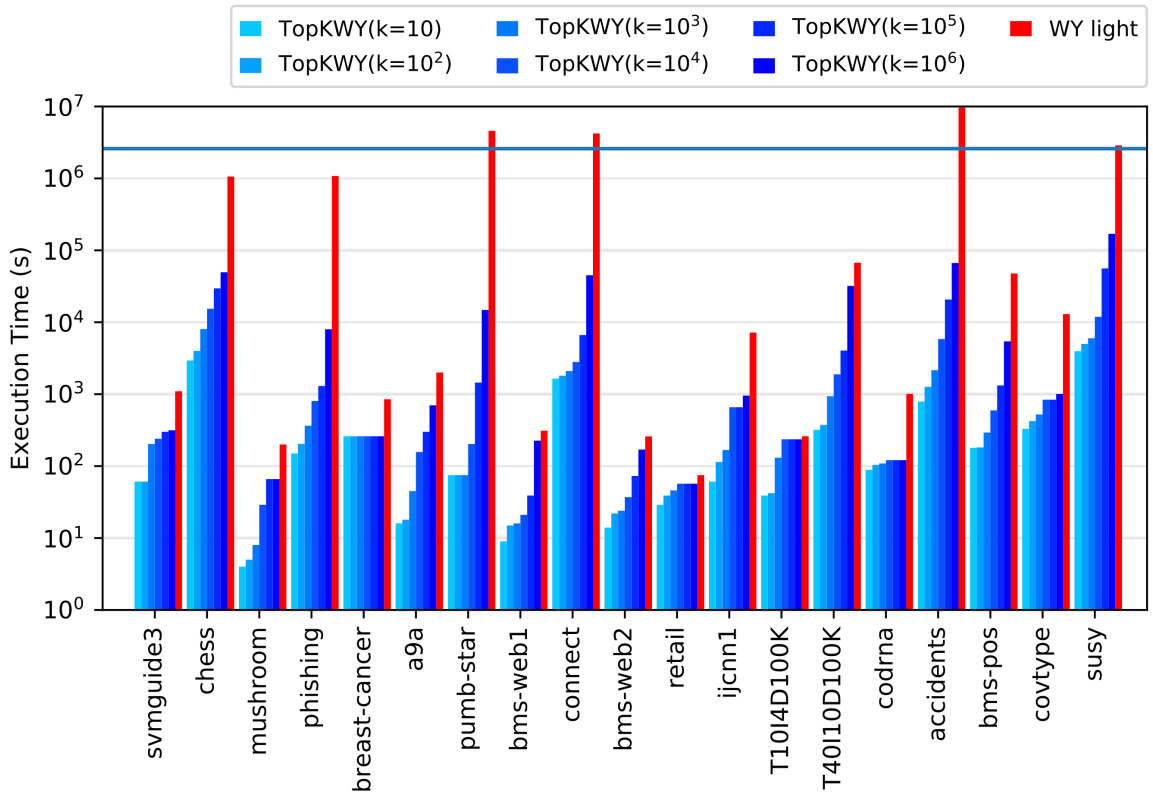


Figure 3-1: Running time for TopKWY (with various values of k) and WYlight. The blue horizontal line corresponds to 1 month of computation.

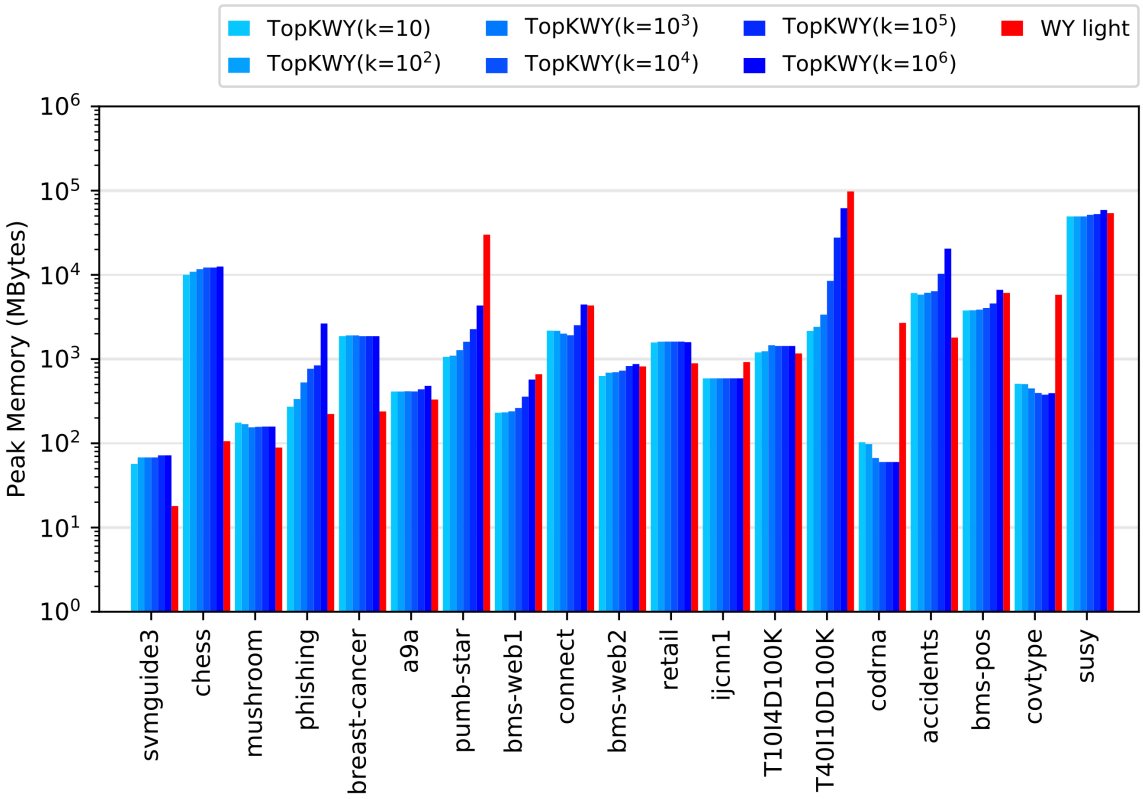


Figure 3-2: Peak memory for TopKWY (with various values of k) and WYlight.

at least one order of magnitude, and for 6 datasets WYlight requires > 11 days while TopKWY identifies up to the 10^6 most significant patterns within one day and the 10^4 most significant ones in few hours. Even for the datasets where TopKWY identifies all significant patterns, producing the same patterns as WYlight, TopKWY is always faster than WYlight, with up to one order of magnitude speed-up in some cases. This shows that TopKWY is an effective tool to identify all significant patterns whenever possible and enables the analysis of significant patterns when their number is extremely high. For datasets in which the number of significant patterns is $> 10^6$ we ran TopKWY with $k = \infty$ to compare its strategy for finding the corrected significance threshold for *all* significant patterns with the one used by WYlight. The runtime of TopKWY is always lower than the runtime of WYlight by at least 20%, with a significant speed-up in some case (e.g., for chess, TopKWY terminates in 2 days, while WYlight needs more than 10 days). These results show that TopKWY outperform the state-of-the-art even for this task.

Figure 3-2 compares the peak memory required by TopKWY and WYlight. Given the *best first* strategy employed by TopKWY, we expected its memory requirement could be higher than WYlight, that follows a depth first strategy. Interestingly, only in three cases TopKWY required 1 order of magnitude more memory than WYlight and in both such cases the requirements are reasonable (≤ 20 GBs) for current machines. However, in such cases WYlight required > 11 days to complete, while TopKWY terminated in < 1 day, showing that, by using a reasonably larger amount of memory than WYlight, TopKWY renders the identification of significant patterns feasible. In all other cases the memory requirement of TopKWY is either the same or within few GBs of WYlight. For some datasets TopKWY requires significantly less memory than WYlight: surprisingly this happens for datasets (cod-rna, covtypes) on which TopKWY reports the same significant patterns as WYlight (i.e., *all* significant patterns). In some cases, memory usage decreases slightly when k increases, due to our dynamical allocation of the p-values lookup table that may require less space when the minimum support decreases.

We investigated the impact of our implementation choices on the memory requirement of TopKWY (Figure 3-3). We compared the space required to store the permuted labels on all the nodes of the PatriciaTrie used by TopKWY (see Section 3.6.1) with the space required by storing the permuted labels for each transaction (as done for example by WYlight). Since TopKWY stores, for each node of the Patricia Trie, a list of m values (i.e., the number of transactions with minority label among the ones sharing the prefix corresponding to the node), one transaction may have more than m values associated to its nodes. In most cases the space required by the two methods is essentially the same, but in three cases the use of the Patricia Trie corresponds to a significant reduction in the memory used. In particular, these three cases are for datasets in which TopKWY identifies all the significant patterns using less memory than WYlight, providing strong evidence of the importance of our encoding of the permuted class labels.

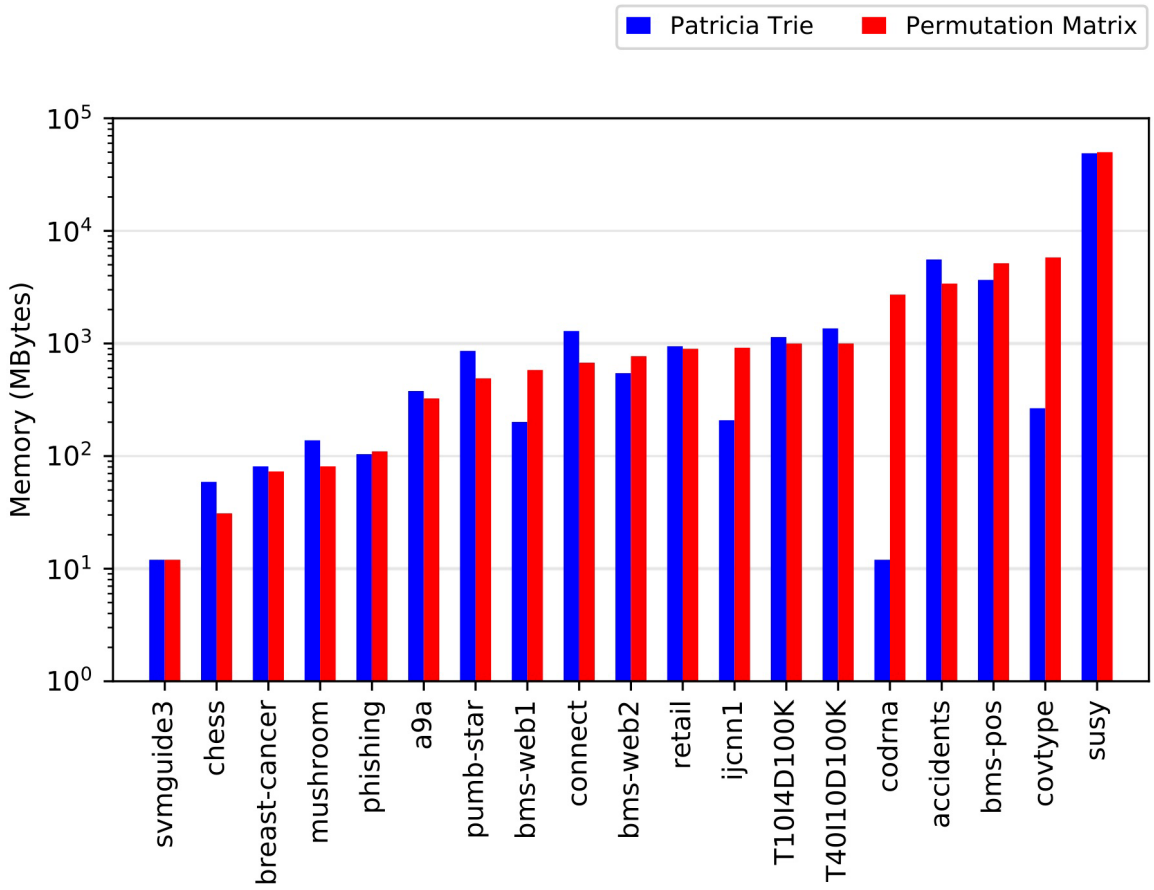


Figure 3-3: Memory requirement for permuted class labels using PatriciaTrie and permutation matrix.

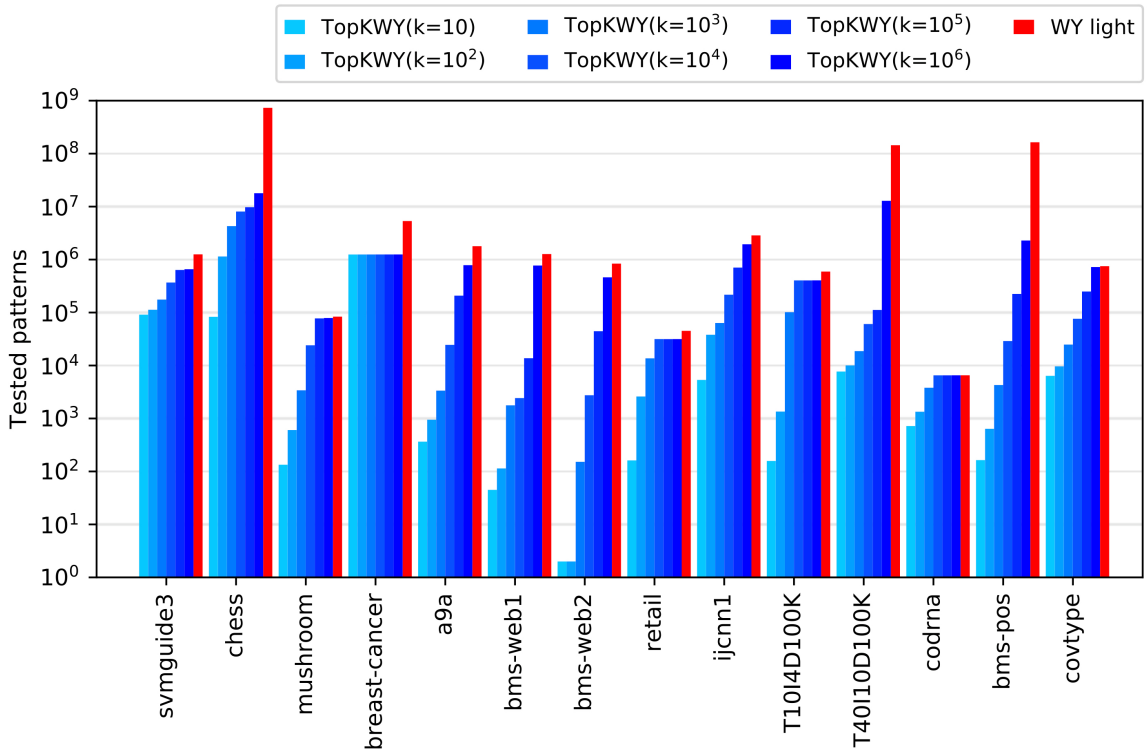


Figure 3-4: Comparison between the number of tested patterns on the permuted datasets by TopKWY and WYlight.

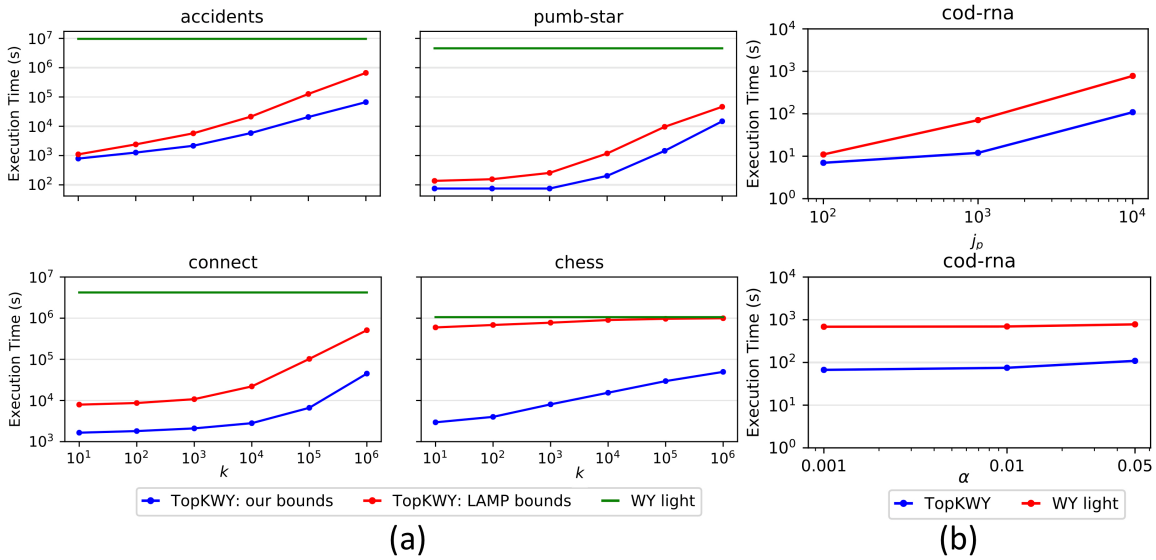


Figure 3-5: (a) Comparison between the running time of WYlight and the running time of TopKWY using our improved bound $\psi'(\cdot)$ and the LAMP bound $\hat{\psi}(\cdot)$. (b) Running time for different values of α and m.

We compared the exploration strategies used by TopKWY and by WYlight by recording the number of patterns they test (Figure 3-4), restricting to datasets in which WYlight terminates. In all cases, TopKWY tests a lower number of patterns than WYlight, with differences of almost two orders of magnitude for some datasets. This shows the effectiveness of our exploration strategy and of our novel bounds $\psi'(\cdot)$ (see Section 3.4) on reducing the number of tests to perform.

We then directly investigated the impact of our novel bounds on the runtime of TopKWY. We compared the running time of WYlight with the running time of two variants of TopKWY: one using our improved bound $\psi'(\cdot)$ and one using the LAMP bound $\hat{\psi}(\cdot)$ (i.e., the same bound used by WYlight). The results for some representative datasets are in Figure 3-5(a). The results for the other datasets are similar. We observed that, for all datasets other than chess, the exploration strategy employed by TopKWY to extract only the Top-k Significant Patterns already provides a substantial (up to more than one order of magnitude) improvement in the running time of TopKWY with respect to WYlight, even using the same LAMP bounds. When our novel bound $\psi'(\cdot)$ is used in TopKWY we observe additional speed-ups, for a total up to more than two orders of magnitude. Therefore, the reduction in the number of patterns that need to be tested on the permuted datasets, obtained by the exploration strategy of TopKWY and our improved bound, is a crucial component for the performance of TopKWY.

Finally, we assessed the impact of α and m on the running time of TopKWY and WYlight on two representative datasets, cod-rna and accidents, which are rep-

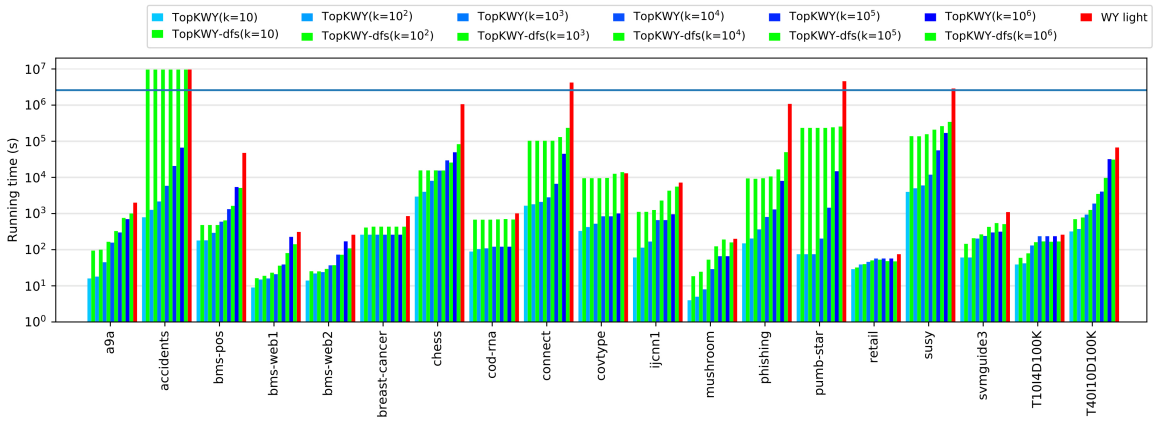


Figure 3-6: Running time for TopKWY, TopKWY-dfs (with various values of k) and WYlight. The blue horizontal lines corresponds to 1 month of computation.

resentative for the two scenarios of a small number of significant patterns (cod-rna) and of a large number of significant patterns (accidents). In these experiments we fixed $k = 10^4$. Figure 3-5(b) reports the results for cod-rna. Results for accidents are not reported since the running time of accidents remained essentially the same for all values of α and m. This means that for accidents using the bounds introduced in Section 3.4 the computational effort is dominated by the pattern space exploration (and not the evaluation of the permuted datasets): considering only the Top-k Significant Patterns is therefore crucial to analyze such dataset. For cod-rna, we observe that varying α has some but small impact on the runtime of both methods while there is a linear dependence of the running time of WYlight on m and a similar but less pronounced dependence of TopKWY. In all cases, TopKWY is faster than WYlight (for accidents WYlight does not terminate within 1 month) showing the efficiency of TopKWY for different ranges of the α and m parameters.

Comparison between Best First and Depth First strategies. We investigate the impact of the best first strategy adopted by TopKWY on the computational performances of the mining tasks. To do so, we compared TopKWY with TopKWY-dfs, the variant of TopKWY, which explores patterns in depth first order (see Section 3.5.3). We ran TopKWY-dfs on the same set of experiments described in Section 3.7.3. Figure 3-6 shows the running times of TopKWY, TopKWY-dfs, and WYlight. We can clearly see that, for 9 datasets out of 19, there is a significant difference in the running times of TopKWY and TopKWY-dfs: this means that, in particular for smaller values of k, the exploration strategy is a critical component of TopKWY. For accidents, one of the most challenging dataset to analyze, both TopKWY-dfs and WYlight can not complete their execution in less than one month, even for $k = 10$. Therefore, for such dataset the best first strategy adopted by TopKWY is crucial.

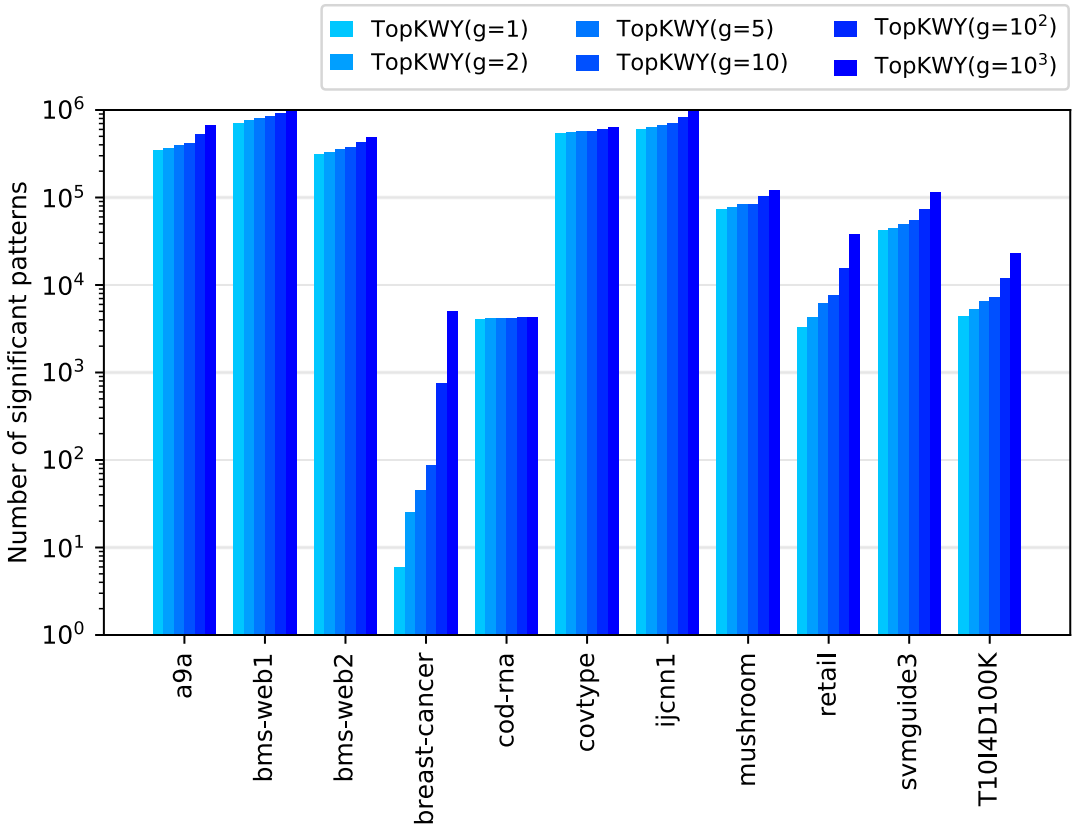


Figure 3-7: Number of results found using TopKWY when controlling the g-$FWER$, for various values of g, $m = 10^4$, $k = 10^6$, and $\alpha = 0.05$.

Results for g-$FWER$ We investigate the increase in statistical power of TopKWY when controlling the generalized-Family-Wise Error Rate (g-$FWER$) by analyzing datasets described in Section 3.7.3 having less than 10^6 results for $\alpha = 0.05$ when controlling the $FWER$. As we can see in Figure 3-7, for the breast-cancer dataset the number of significant patterns increased by more then two orders of magnitude as the value of g increases (i.e., when more false positives are allowed). Figure 3-8 show the running time of TopKWY when controlling the g-$FWER$ at different values of g. As expected, the required time slightly increases but it stays practical for all datasets. We do not compare with other methods since TopKWY is the first algorithm to discover significant patterns with a rigorous control on the g-$FWER$.

In Table 3.3.a we show the computed values of $g^\star$ by TopKWY* (see Section 3.5.1 for its definition) on the breast-cancer dataset for $k \in \{10, 10^2, 10^3, 10^4\}$. We can see that TopKWY* is able to provide informative estimates of the quality of the reported set of k most significant patterns, in terms of the minimum g such that the g-$FWER(p^k)$ is $\leq \alpha$, without the need of fixing g a-priori. In Table 3.3.b we show the number of results found using TopKWY-ζ for $k = 10^4$ on the breast-cancer dataset,

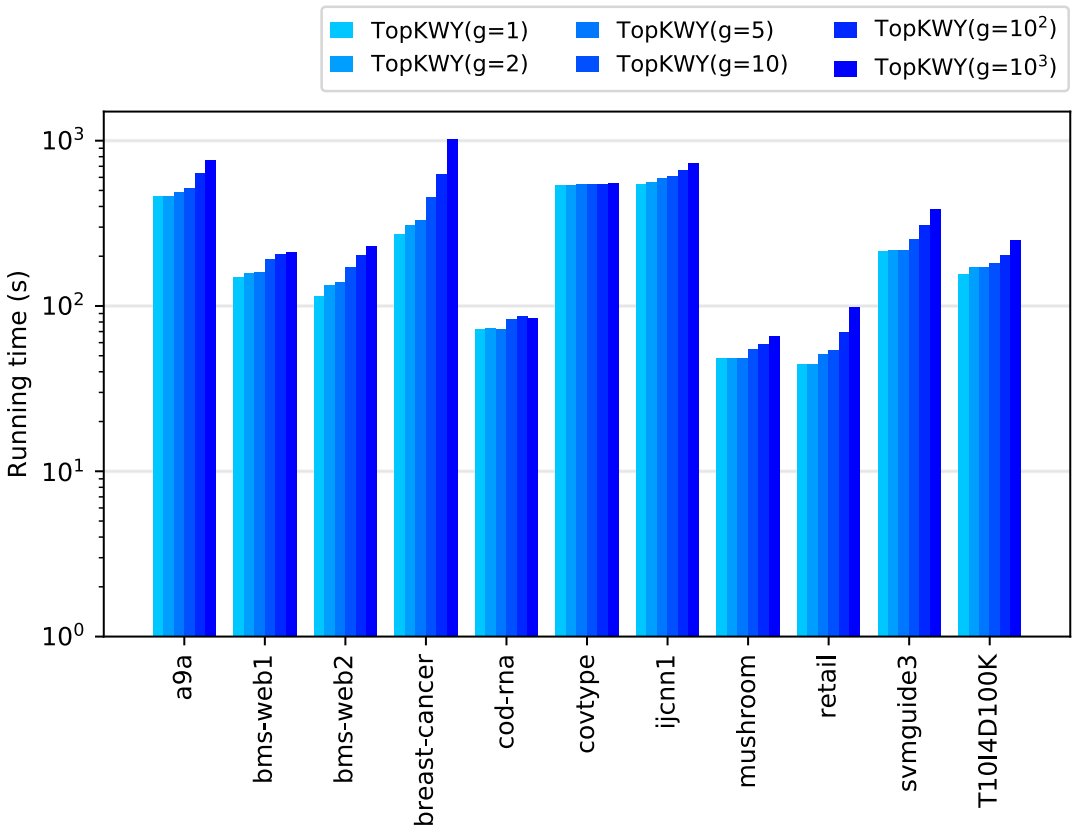


Figure 3-8: Running time for TopKWY when controlling the g-$FWER$, for various values of g, $m = 10^4$, $k = 10^6$, and $\alpha = 0.05$.

Table 3.3: a) Values of $g^\star$ (second row of the Table) computed by TopKWY* for different values of k (first row of the Table) on breast-cancer dataset with $m = 10^4$ and $\alpha = 0.05$. b) Number of results $|\mathcal{V}|$ (second row of the Table) found by TopKWY-ζ on breast-cancer dataset with $m = 10^4$, $\alpha = 0.05$, and $k = 10^4$, for different values of ζ (first row of the Table).

a)

k	10	10^2	10^3	10^4
$g^\star$	2	11	163	2396

b)

ζ	0.01	0.05	0.1	0.25
$\lvert\mathcal{V}\rvert$	0	24	624	10^4

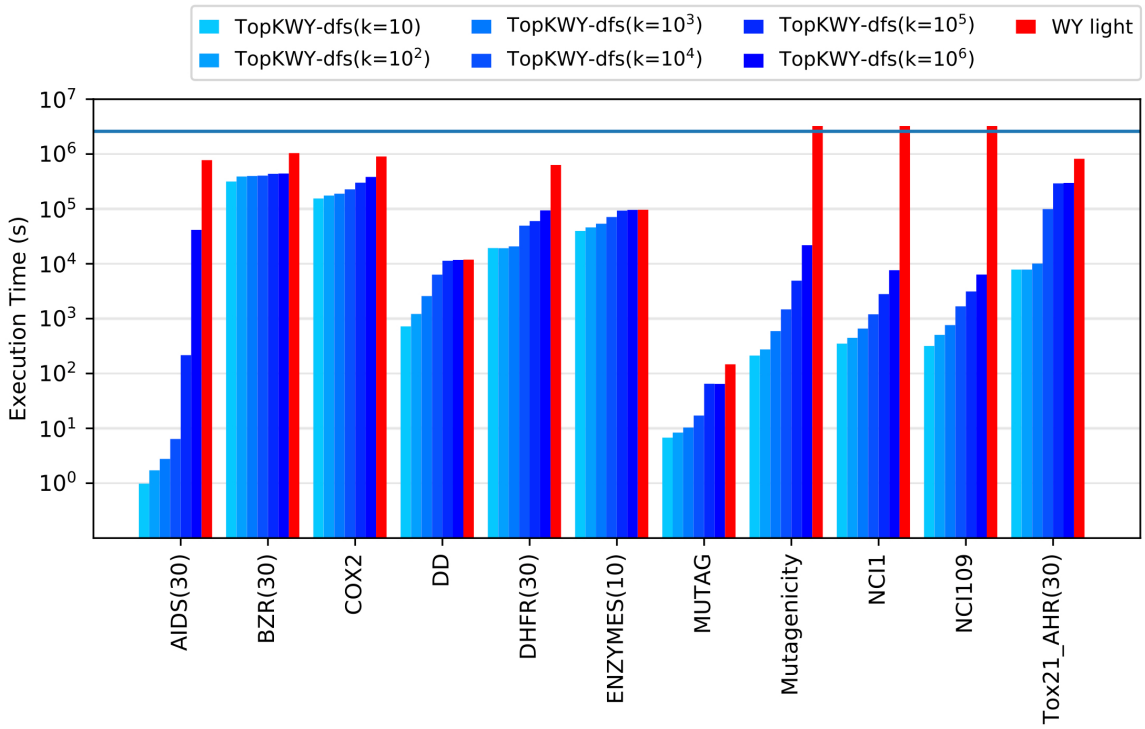


Figure 3-9: Running time for TopKWY-dfs (with various values of k) and WYlight for significant subgraph mining. The blue horizontal line corresponds to 1 month of computation.

varying $\zeta \in \{0.01, 0.05, 0.1, 0.25\}$. From these results we can see that TopKWY-ζ is a very flexible tool to discover significant patterns with bounds on both the output size and the maximum ratio of false discoveries, providing improved statistical power in situations where the number of significant patterns when controlling the $FWER$ is very low. (We do not show the running times for TopKWY* and TopKWY-ζ since those are very similar to the ones reported in Figure 3-8.)

Subgraphs Mining

We ran TopKWY-dfs on the datasets described in Section 3.7.2. Table 3.2 reports the number of significant patterns for $\alpha = 0.05$ in the datasets we considered, obtained by running TopKWY-dfs (with $k = +\infty$) or WYlight. For some datasets we stopped the computation after 1 month, so only a lower bound is available. In most cases, the number of significant patterns is extremely large: for 6 out of 11 datasets there are $> 5 \times 10^5$ significant patterns and in 5 datasets there are $> 10^6$ significant patterns. This shows that a direct way to limit the number of significant patterns in output is required for subgraphs mining as well.

We then compared the running time and memory requirement of TopKWY-

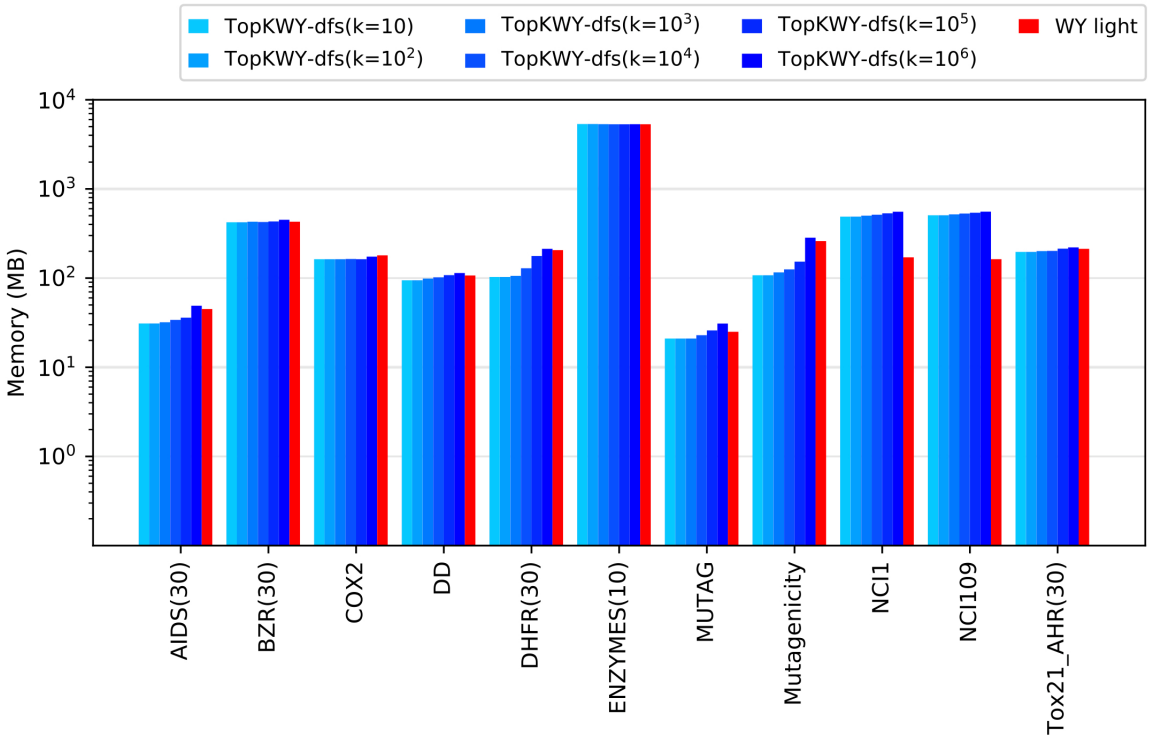


Figure 3-10: Memory usage for TopKWY-dfs (with various values of k) and WYlight for significant subgraph mining.

dfs and of WYlight. Figure 3-9 compares the running times of TopKWY-dfs and WYlight. As for itemsets mining, when the number of significant patterns is lower than k, TopKWY-dfs finds all of them, obtaining the same output as WYlight. We can see that TopKWY-dfs is, in all cases, faster than WYlight: for 9 datasets out of 11 and for $k = 10^6$, TopKWY-dfs improves the running time by a factor at least 2. It is interesting to note that for 5 datasets the number of significant results is $< 10^6$, therefore TopKWY-dfs is faster even if its output is the same of WYlight. For 3 datasets the running time is reduced by more than two orders of magnitude, and WYlight is not able to terminate in less than 1 month. We can observe that these three datasets contains more than 10^6 significant results; this clearly shows that focusing on the most significant patterns leads to significant computational advantages and enables the analysis of such datasets.

Figure 3-10 compares the memory usage of TopKWY-dfs and WYlight. Both algorithms are very memory efficient and, while TopKWY-dfs usually requires more memory than WYlight, the difference is small: for 7 of the 8 datasets where WYlight terminates, TopKWY-dfs never requires more than 8% of the memory of WYlight, and 23% in the case of MUTAG, where the difference is of few MBs. (For the three datasets were WYlight does not terminate, we only report a lower bound to its memory usage.)

Chapter 4

Significant Pattern Mining with Unconditional Testing

4.1 Introduction

The significance of a pattern in Significant Pattern Mining 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 (Fisher, 1922). Fisher's test is a *conditional test*: it assumes that the data generating process only produces datasets in which both the number of transactions with the same binary label *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 (Barnard, 1945) assume 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 labels considering also scenarios (i.e., datasets) where the frequency of the pattern is different from what is observed in the real data. 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) properties 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.

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 popularity of Barnard's test was hindered partially by Fisher's criticism, and partially by the excessive computational cost required by naïve implementations of

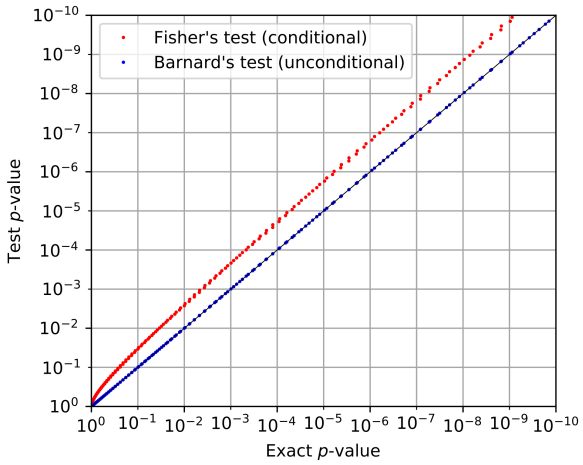


Figure 4-1: p-values from Fisher's test (conditional) and Barnard's tests (unconditional) vs. exact p-values under the unconditional null hypothesis. p-values of both tests are displayed for all contingency tables with $n = 10^4$, $n_1/n = 0.25$, $\mathsf{f}(\mathcal{P}) = 0.1$, and all $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}$ (see Section 4.2 for parameters' definitions) and exact p-value $\geq 10^{-10}$. The diagonal black line corresponds to a test p-value equal to the exact p-value. The p-values from Fisher's test are *smaller* than the exact p-values for many tables (the values on the axes *decrease* toward the right and upwards) and may lead to different conclusions.

the test. We do not enter the debate on which of the two tests to use (Mehta and Senchaudhuri, 2003; Boschloo, 1970; Yates, 1984; Berger, 1994; Choi et al., 2015), 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 Hämäläinen (2016), who studied computationally efficient upper bounds to the p-value of Fisher's test.

The difference between using conditional and unconditional tests is usually small when testing the significance of a *single* pattern $\mathcal{P}$: in this case one can flag $\mathcal{P}$ 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 $\mathcal{P}$ as significant when it is not) is bounded by α. 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, as we discussed in Section 2.2. All the several methods to correct for this issue we introduced in Section 2.2.2 require patterns to have very small p-values in order to be flagged as significant. For patterns with very small p-values, conditional tests and unconditional tests display strikingly different p-values (Figure 4-1), even if the size of the data is not small (i.e., when $n = 10^4$). This discrepancy highlights the difference between imposing conditional and unconditional assumptions, and shows that different conclusions on the significance of the patterns

are very likely to be concluded. Vandin et al. (2015) observed similar discrepancies for the log-rank test.

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

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 formulation of an 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 for 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 may find applications in other Significant Pattern Mining problems.

- 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 (Terada et al., 2013a). 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 (Riondato and Upfal, 2015), rather than having to expensively compute a different confidence interval for each pattern. To the best of our knowledge SPuManTE is the first algorithm in which such uniform bounds have been used, and we believe that this approach could be applied to other methods for Significant Pattern Mining.

- We evaluate SPuManTE on real datasets and compare its performance with the state-of-the-art method LAMP (Terada et al., 2013a), 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 the 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.

4.2 Preliminaries

In this Section we refresh the notation introduced in Section 2.1. Let $\mathcal{I}$ be an alphabet of ordered *items*, and let $\{\ell^0, \ell^1\}$ be two *(class) labels*. A *dataset* $\mathcal{D} = \{(t_1, \ell_1), (t_2, \ell_2), \ldots, (t_n, \ell_n)\}$ 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 $\mathcal{D}^0$ and $\mathcal{D}^1$, where $\mathcal{D}^i$ 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 $n_i = |\mathcal{D}^i|$, with $n_1 + n_0 = n$.

A *pattern* (or itemset) $\mathcal{P}$ is a set of items, $\mathcal{P} \subseteq \mathcal{I}$. We say that $\mathcal{P}$ *appears* in a transaction t if $\mathcal{P} \subseteq t$, and say that t *contains* $\mathcal{P}$. The *support* $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^i}$ (resp. *frequency* $\mathsf{f}_i(\mathcal{P})$) *of* $\mathcal{P}$ *in* $\mathcal{D}^i$ is the *number* (resp. *fraction*) of transactions in $\mathcal{D}^i$ that contain $\mathcal{P}$, for $i \in \{0, 1\}$. We denote as $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}$ (resp. $\mathsf{f}(\mathcal{P})$) the number (resp. fraction) of transactions in the pairs of $\mathcal{D}$ that contain $\mathcal{P}$. Thus, $\mathsf{f}_i(\mathcal{P}) = \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^i}/n_i$, $i \in \{0, 1\}$, and $\mathsf{f}(\mathcal{P}) = \mathsf{z}^{\mathcal{P}}_{\mathcal{D}}/n$. For any pattern $\mathcal{P}$, these quantities (and their complements) are summarized in a 2×2 contingency table such as the one in Table 2.1. We conveniently define the quantities $\check{\mathsf{z}}_{i,x} = \max\{0, n_i - (n - x)\}$ and $\hat{\mathsf{z}}_{i,x} = \min\{n_i, x\}$, that define the range $[\check{\mathsf{z}}_{i,x}, \hat{\mathsf{z}}_{i,x}]$ of the admissible values of $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^i}$ for any $\mathcal{P}$ with $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}} = x$.

In Significant Pattern Mining, the dataset $\mathcal{D}$ is assumed to be the outcome of a stochastic process that generates sets of pairs (t, λ). Different assumptions can be made on this process (details in Sections 4.2.1 and 4.2.2). Independently on the assumptions, for any pattern $\mathcal{P} \subseteq \mathcal{I}$, we let $\pi_{\mathcal{P},i}$ be the probability that a pair (t, λ) generated by the process is such that $\mathcal{P} \subseteq t$ 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 $\pi_{\mathcal{P},0} \neq \pi_{\mathcal{P},1}$. Given a single pattern $\mathcal{P}$, assessing the statistical significance of an association corresponds to using the *observed* contingency table for $\mathcal{P}$ to evaluate whether it *supports* the *null hypothesis* $H_{\mathcal{P}} : \pi_{\mathcal{P},0} = \pi_{\mathcal{P},1}$, i.e., whether the observed data $\mathcal{D}$ is likely to have been generated from a process satisfying $H_{\mathcal{P}}$.

All available information about $\mathcal{P}$ is contained in the contingency table, thus we cannot be *deterministically certain* in our assessment of the significance of the association of $\mathcal{P}$ 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 $\mathcal{P}$ is supported by the observed contingency table for $\mathcal{P}$ involves computing a *p-value* $p_{\mathcal{P}}$. 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 $\mathcal{P}$ 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 with each of them depend on the *conditions* imposed on the data generation process (we discuss this aspect in Sections 4.2.1 and 4.2.2).

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

4.2.1 Conditional Testing

Let's focus for now on a single pattern $\mathcal{P} \subseteq \mathcal{I}$. A set of conditions commonly imposed on the data generating process, made for example by the widely-used Fisher's exact test (Fisher, 1922) we defined in Section 2.2.1, is to *condition on all values in the marginals* (i.e., in the bottom row and rightmost column) of the observed contingency table for $\mathcal{P}$: the space of possible contingency tables for $\mathcal{P}$ contains all and only those with the same values for n, n_1 (thus n_0), and $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}$, as in the observed one.

Under these conditions and assuming that the null hypothesis $H_{\mathcal{P}}$ is true, the quantity $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}$ follows a *hypergeometric distribution*: given $x = \mathsf{z}^{\mathcal{P}}_{\mathcal{D}}$ and $a \in [\check{\mathsf{z}}_{1,x}, \hat{\mathsf{z}}_{1,x}]$, the probability $\mathsf{h}(a, \mathcal{P}, \mathcal{D})$ of observing a contingency table for $\mathcal{P}$ where $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1} = a$ is

$$\mathsf{h}(a, \mathcal{P}, \mathcal{D}) = \binom{n_1}{a}\binom{n_0}{x-a} \Big/ \binom{n}{x} \quad .$$

Let b be the value for $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}$ in the observed contingency table for $\mathcal{P}$, the p-value $p^{\mathsf{F}}_{H_{\mathcal{P}}}(\mathcal{D})$ of $\mathcal{P}$ on $\mathcal{D}$ is then

$$p^{\mathsf{F}}_{H_{\mathcal{P}}}(\mathcal{D}) = \sum_{a:\mathsf{h}(a,\mathcal{P},\mathcal{D}) \leq \mathsf{h}(b,\mathcal{P},\mathcal{D})} \mathsf{h}(a, \mathcal{P}, \mathcal{D}) \quad . \tag{4.1}$$

Drawbacks Imposing on the generative process the conditions we just described may be reasonable when only considering a single pattern $\mathcal{P}$. 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* $\mathcal{P}$ of the $2^{|\mathcal{I}|}$ possible patterns, the generative process only generates contingency tables for $\mathcal{P}$ with the observed value of $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}$. Imposing such a large number of conditions seems excessively restrictive.

4.2.2 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 quantities are characteristics of the observed dataset, and much more readily accessible than 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 (Barnard, 1945), are known as *unconditional tests*. For a pattern $\mathcal{P}$, 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 $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}$ 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 $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}$ for each pattern $\mathcal{P}$ is unknown and must be obtained by running a pattern mining algorithm on the dataset.

To define the p-value for $\mathcal{P}$ we need to first introduce the concept of the *nuisance*

parameter $\pi \in [0,1]$. The nuisance parameter is the *assumed value* for $\pi_{\mathcal{P},0}$ and $\pi_{\mathcal{P},1}$ under the null hypothesis that these quantities are equal.

Under the above conditions, and assuming that the null hypothesis $H_{\mathcal{P}}$ is true, and a fixed, known value for the nuisance parameter π, the probability, given by the function $\mathsf{b}(x, a \mid \pi)$, of observing a contingency table for $\mathcal{P}$ with $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}} = x$ and $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1} = a$, for $x \in [0, n]$, and $a \in [\check{\mathsf{z}}_{1,x}, \hat{\mathsf{z}}_{1,x}]$, is

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

For $y \in [0, n]$, $b \in [\check{\mathsf{z}}_{1,y}, \hat{\mathsf{z}}_{1,y}]$, and $\pi \in (0, 1)$, define $T(y, b, \pi)$ as the set of "more extreme outcomes", composed by pairs (x, a) such that $\mathsf{b}(x, a \mid \pi) \leq \mathsf{b}(y, b \mid \pi)$ and define the function

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

The value $\phi(y, b, \pi)$ is the probability, under the null hypothesis $\pi_{\mathcal{P},0} = \pi_{\mathcal{P},1}$ and for a fixed value π of the nuisance parameter, to observe a contingency table as or more extreme than the one with $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}} = y$, $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1} = b$, and $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0} = y - b$ (see Table 2.1).

To obtain the actual p-value for a pattern $\mathcal{P}$ or an upper bound to it (sufficient to perform the test), it is necessary to eliminate the dependency on nuisance parameter π. For example, Barnard's test (Barnard, 1945) uses as upper bound $p^{\mathsf{B}}_{\mathcal{P}}$ the maximum of $\phi(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}, \pi)$ over all values of $\pi \in [0, 1]$:

$$p^{\mathsf{B}}_{\mathcal{P}} = \max\left\{\phi(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}, \pi), 0 \leq \pi \leq 1\right\} \ . \tag{4.2}$$

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 Section 4.3.1).

4.2.3 Multiple Hypothesis Testing

When a single pattern $\mathcal{P}$ 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 $\mathcal{P}$ 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 grows with h. Therefore, we are interested in deriving an appropriate significance threshold $\delta^\star$ such that the Family-Wise Error Rate ($FWER$) is controlled below a level α.

To identify a suitable significance threhsold $\delta^\star$ to control the $FWER$, in SPuMaNTE we employ a strategy similar to the one in LAMP (see Section 2.3), but using our unconditional test (see Section 4.3.1), thus requiring the development of an efficiently computable lower bound to the minimum attainable p-value for a pattern

$\mathcal{P}$ using such test (Section 4.3.2).

4.3 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. For the ease of exposition, some of the proofs are postponed to Section 4.5.

4.3.1 The UT test

Let $\mathcal{P}$ be a pattern of which we are assessing the association with the labels. Our Unconditional Test UT assumes to know two confidence intervals $C_0(\mathcal{P})$ and $C_1(\mathcal{P})$, for $\pi_{\mathcal{P},0}$ and $\pi_{\mathcal{P},1}$, respectively, s.t. the event $E_{\mathcal{P}} =$ "$\pi_{\mathcal{P},0} \in C_0(\mathcal{P})$ and $\pi_{\mathcal{P},1} \in C_1(\mathcal{P})$" holds with probability $1-\gamma$ (over the randomness in the data generation process). This idea is similar to the approach by Berger (1996); however, we discuss in Section 4.3.2 how such confidence intervals can be obtained *simultaneously* for all patterns $\mathcal{P}$, instead of potentially looser confidence intervals built *individually*. Let the interval $C(\mathcal{P})$ be $C(\mathcal{P}) = C_0(\mathcal{P}) \cap C_1(\mathcal{P})$. We define the p-value $p_{\mathcal{P}}$ conditioned on the event $E_{\mathcal{P}}$ as

$$p_{\mathcal{P}} = \begin{cases} 0 & \text{if } C(\mathcal{P}) = \emptyset \\ \max\left\{\phi(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}, \pi), \pi \in C(\mathcal{P})\right\} & \text{otherwise.} \end{cases} .$$

This p-value should be compared with the one from (4.2). The p-value $p_{\mathcal{P}}$ 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 $E_{\mathcal{P}}$. This conditioning is entirely different than the conditioning made by conditional tests such as Fisher's (Fisher, 1922), which conditions on the *observed* supports of the patterns.

Given a fixed threshold α, UT flags $\mathcal{P}$ as significant *iff* $p_{\mathcal{P}} \leq \alpha - \gamma$. The following property holds.

Theorem 4.3.1. *Let $\mathcal{P}$ be a fixed pattern. The probability that UT flags $\mathcal{P}$ as significant when it is not is at most α.*

Proof. Let F be the event "UT flags $\mathcal{P}$ as significant when it is not", which corresponds to a false discovery. It holds

$$\Pr(F) = \Pr(F \mid E_{\mathcal{P}}) \Pr(E_{\mathcal{P}}) + \Pr(F \mid \bar{E}_{\mathcal{P}}) \Pr(\bar{E}_{\mathcal{P}}) \leq \Pr(F \mid E_{\mathcal{P}}) + \Pr(\bar{E}_{\mathcal{P}}),$$

where $\bar{E}_{\mathcal{P}}$ denotes the event complementary to $E_{\mathcal{P}}$. By the definition of confidence interval, $\Pr(\bar{E}_{\mathcal{P}}) \leq \gamma$, while $\Pr(F \mid E_{\mathcal{P}}) \leq \alpha - \gamma$ since when $E_{\mathcal{P}}$ holds we are using

the standard hypothesis testing framework with significance threshold $\alpha - \gamma$ and the p-value $p_{\mathcal{P}}$. □

An upper bound to the p-value

The exact computation of $p_{\mathcal{P}}$ when $C(\mathcal{P}) \neq \emptyset$ requires an expensive search over the values of $\pi \in C(\mathcal{P})$. For the purpose of testing the significance of a pattern and ensuring that Theorem 4.3.1 still holds, only an efficient-to-compute *upper bound* to the p-value is needed. We prove the following.

Theorem 4.3.2. $p_{\mathcal{P}} \leq \mathsf{b}\left(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_{\mathcal{D}^1}^{\mathcal{P}} \mid \mathsf{f}(\mathcal{P})\right)(n_0+1)(n_1+1)$.

Most importantly, the upper bound $\hat{p}_{\mathcal{P}}$

$$\hat{p}_{\mathcal{P}} = \mathsf{b}\left(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_{\mathcal{D}^1}^{\mathcal{P}} \mid \mathsf{f}(\mathcal{P})\right)(n_0+1)(n_1+1)$$

can be computed in $\mathcal{O}(1)$ time.

A lower bound to the p-value

We now show a lower bound to $p_{\mathcal{P}}$. More than being just of theoretical interest, this lower bound is the starting point to derive, in Section 4.3.2, an efficiently-computable, monotonically-non-increasing lower bound to the minimum attainable p-value for a pattern $\mathcal{P}$ (required to compute the corrected significance threshold in a way similar to what is done by LAMP (Terada et al., 2013a)).

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

Lemma 4.3.3. *If $C(\mathcal{P}) \neq \emptyset$, then $p_{\mathcal{P}} \geq \phi(y, b, \pi)$ for any $\pi \in C(\mathcal{P})$.*

Proof. The statement follows from the definition of $p_{\mathcal{P}}$, that is the maximum over $\pi \in C(\mathcal{P})$ of the r.h.s. □

Theorem 4.3.3 states that any $\pi \in C(\mathcal{P})$ allows to obtain a lower bound to $p_{\mathcal{P}}$, so we choose $\pi = \mathsf{f}(\mathcal{P})$, and define the lower bound $\check{p}_{\mathcal{P}}$ to $p_{\mathcal{P}}$ as

$$\check{p}_{\mathcal{P}} = \phi(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_{\mathcal{D}^1}^{\mathcal{P}}, \mathsf{f}(\mathcal{P})) \ . \tag{4.3}$$

Our choice for π is driven by the objective of maximizing the number of contingency tables in $T(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_{\mathcal{D}^1}^{\mathcal{P}}, \pi)$, which we try heuristically to achieve by maximizing $\mathsf{b}(\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}, \mathsf{z}_{\mathcal{D}^1}^{\mathcal{P}} \mid \pi)$, which is straightforward as shown in the following result.

Proposition 4.3.4. *It holds* $\arg\max_{\pi} \{\mathsf{b}(x, a \mid \pi)\} = x/n$.

We show in Section 4.4.1 that $\check{p}_{\mathcal{P}}$ provides a tight lower bound to $p_{\mathcal{P}}$.

4.3.2 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.

Simultaneous confidence intervals

For each tested pattern $\mathcal{P}$, UT requires confidence intervals for $\pi_{\mathcal{P},0}$ and $\pi_{\mathcal{P},1}$. 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 (Riondato and Upfal, 2015) to derive confidence intervals for the quantities $\pi_{\mathcal{P},0}$ and $\pi_{\mathcal{P},1}$ 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 (2015), called AMIRA, to obtain a value $\varepsilon \in (0,1)$ with the following property. Given this ε, define, for each pattern $\mathcal{P}$, the intervals $C_0(\mathcal{P})$ and $C_1(\mathcal{P})$ as

$$C_0(\mathcal{P}) := \left[\mathsf{f}_0(\mathcal{P}) - \varepsilon\frac{n}{n_0}, \mathsf{f}_0(\mathcal{P}) + \varepsilon\frac{n}{n_0}\right],$$
$$C_1(\mathcal{P}) := \left[\mathsf{f}_1(\mathcal{P}) - \varepsilon\frac{n}{n_1}, \mathsf{f}_1(\mathcal{P}) + \varepsilon\frac{n}{n_1}\right],$$

and define the event $E_{\mathcal{P},\varepsilon}$ = "$\pi_{\mathcal{P},0} \in C_0(\mathcal{P})$ *and* $\pi_{\mathcal{P},1} \in C_1(\mathcal{P})$". Consider the event $E_\varepsilon = \bigcap_{\mathcal{P}\subseteq\mathcal{I}} E_{\mathcal{P},\varepsilon}$. 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).

Lower bound to the minimum attainable 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 $\mathcal{P}$ for which $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}} \leq x$, $x \in [0,n]$. Having this lower bound is crucial to prune the search space of testable patterns.

Our bounds crucially hinges on the following results, under the ongoing assumption $n_1 \leq n_0$.

Theorem 4.3.5. *Let* $C_0(\mathcal{P}) \cap C_1(\mathcal{P}) = C(\mathcal{P}) \neq \emptyset$. *Then* $\mathsf{f}(\mathcal{P}) \in C(\mathcal{P})$.

For $x \in [0, n]$ let

$$Q(x) = \Bigg\{ (r, w), r \in [\check{\mathsf{z}}_{0,y}, \hat{\mathsf{z}}_{0,y}], w \in [\check{\mathsf{z}}_{1,y}, \hat{\mathsf{z}}_{1,y}] \text{ s.t. } y = r + w \leq x \\ \wedge \left(\frac{r}{n_0} + \varepsilon \frac{n}{n_0} < \frac{w}{n_1} - \varepsilon \frac{n}{n_1} \vee \frac{w}{n_1} + \varepsilon \frac{n}{n_1} < \frac{r}{n_0} - \varepsilon \frac{n}{n_0} \right) \Bigg\}.$$

Intuitively, $Q(x)$ contains all the pairs (r, w) such that the confidence intervals around r/n_0 and w/n_1 do not intersect. Checking whether $Q(x)$ is non-empty (i.e., there exists *at least* one contingency table with marginal $\leq x$ where the confidence intervals do not intersect) can be done by checking if $Q(x)$ contains either $(\check{\mathsf{z}}_{0,x}, \hat{\mathsf{z}}_{1,x})$ or $(\hat{\mathsf{z}}_{0,x}, \check{\mathsf{z}}_{1,x})$: if $Q(x)$ does not contain either, then it must be empty because the frequencies in each class of less *biased* contingency tables have smaller absolute difference w.r.t. those two cases, therefore it is not possible that the intersection of their confidence intervals is empty. When the intersection of the confidence intervals is not empty, we need the following result to prove our lower bound.

Theorem 4.3.6. $\arg\min_a \{\phi(x, a, \pi)\} = \min\{x, n_1\}$.

Thus, we can define

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

as a lower bound to the minimum attainable $p_{\mathcal{P}}$ for all patterns $\mathcal{P}$ with $\mathsf{z}_{\mathcal{D}}^{\mathcal{P}} = x$. For our purposes, we need a *monotonically non-increasing* lower bound to the minimum attainable p-value, so we define

$$\check{\psi}(x) = \begin{cases} \psi(x) & \text{if } x = 0 \\ \min\{\psi(x), \check{\psi}(x-1)\} & \text{if } x \in [1, n] \end{cases}.$$

SPuMANTE uses $\check{\psi}$ to check whether to mark a pattern $\mathcal{P}$ with $\mathsf{z}_{\mathcal{D}}^{\mathcal{P}} = x$ as untestable when looking for the corrected significance threshold δ. The computation of $\check{\psi}(x)$ can be done efficiently starting from $x = 0$ and increasing x, keeping the values of $\check{\psi}(x)$ in memory.

Efficient computation of ϕ

After having defined $\check{p}_{\mathcal{P}}$ and $\check{\psi}(x)$, we still have to address how to compute them efficiently in the case $C(\mathcal{P}) \neq \emptyset$, 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 $\mathsf{b}(a_0 + a_1, a_1 \mid \pi)$ for all possible pairs $(a_0, a_1) \in [0, n_0] \times [0, n_1]$, leading to the computation of $\Theta(n_0 n_1)$ terms. As we show in Section 4.4.1, even for samples of moderate size this approach is not feasible in reasonable time.

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 $\mathsf{b}(x, a_1 \mid \pi)$ 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 4.3.7. *Let $y \in [0, n]$, $b \in [\check{\mathsf{z}}_{1,y}, \hat{\mathsf{z}}_{1,y}]$, and $\pi \in (0, 1)$. Let*

$$A_1 = \{a_1 : \mathsf{b}\left(a_1 + \lfloor (n_0 + 1)\pi \rceil, a_1 \mid \pi\right) > \mathsf{b}(y, b \mid \pi)\},$$

and define the set

$$A_{0,a_1} = \{a_0 : \mathsf{b}\left(a_1 + a_0, a_1 \mid \pi\right) \leq \mathsf{b}(y, b \mid \pi)\}.$$

Then

$$\sum_{(x,a) \in T(y,b,\pi)} \mathsf{b}(x, a \mid \pi) = \sum_{a_1 \notin A_1} B(a_1, n_1, \pi) + \sum_{a_1 \in A_1} \left(B(a_1, n_1, \pi) \sum_{a_0 \in A_{0,a_1}} B(a_0, n_0, \pi) \right), \tag{4.4}$$

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 Algorithm 2, 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 (4.4) with $\mathcal{O}(1)$ computations 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 (Lentz, 1976), a fast and precise method to evaluate continued fractions. We prove in the Section 4.5 that the time complexity of Algorithm 2 is $\mathcal{O}(\log(n_0) + n + La)$, where $\mathcal{O}(La)$ is the time complexity of Lentz's algorithm.

SPuManTE

Our algorithm SPuManTE outputs a set of significant patterns from $\mathcal{D}$ with $FWER \leq \alpha$. Its pseudocode is presented in Algorithm 3. SPuManTE first obtains (line 1) the maximum deviation ε from AMIRA with parameter γ, so that the event E_ε holds

Algorithm 2: Efficient computation of $\phi(y, b, \pi)$

Input: $y \in [0, n]$, $b \in [\check{\mathsf{z}}_{1,y}, \hat{\mathsf{z}}_{1,y}]$, $\pi \in [0, 1]$.
Output: $\phi(y, b, \pi)$.

```
1  v ← 0
2  z ← b(y, b | π)
3  a'_0 ← ⌊(n_0 + 1)π⌋
4  A_1 ← {a_1 : b(a_1 + a'_0, a_1 | π) > z}
5  forall a_1 ∈ A_1 do
6      a' ← min_{a_0} {a_0 ≤ a'_0 | b(a_0 + a_1, a_1 | π) > z}
7      a'' ← min_{a_0} {a_0 > a'_0 | b(a_0 + a_1, a_1 | π) ≤ z}
8      p' ← C(n_1, a_1) (π)^{a_1} (1 − π)^{(n_1 − a_1)}
9      v ← v + p' (β_π(a', n_0 − a') + β_{1−π}(n_0 + 1 − a'', a''))
10 a' ← max{A_1} + 1
11 a'' ← min{A_1} − 1
12 v ← v + β_π(a' + 1, n_0 + 1 − a') + β_{1−π}(n_0 + 1 − a'', a'')
13 return v
```

with probability $\geq 1 - \gamma$. Then (line 2), SPuMaNTE uses the lower bound $\check{\psi}(x)$ derived in Section 4.3.1 (and computed using Algorithm 2) together with a strategy similar to the one in LAMP (Terada et al., 2013a; Minato et al., 2014) to efficiently derive a corrected significance threshold δ to use in each test while ensuring that the $FWER$ is at most $\alpha - \gamma$. In particular, such strategy initializes the support threshold of *testable patterns* σ_T to 1, and increases it while exploring the closed patterns, reducing the set of testable patterns until the final value of δ is found. Hence, we can incrementally compute the values of $\check{\psi}(x)$ after increasing σ_T by simply comparing $\check{\psi}(\sigma_T - 1)$ to $\psi(\sigma_T)$, therefore only keeping in memory $\check{\psi}(\sigma_T - 1)$ and not the entire function $\check{\psi}(x)$. SPuMaNTE then loops over the testable patterns to test them, to decide whether to flag them as significant or not. It does so by first generating the set of closed patterns $\mathsf{children}(\emptyset)$ that are extensions of the empty pattern $\emptyset$. For every pattern $\mathcal{P}$ of those, it only processes $\mathcal{P}$ if it is testable (therefore if the support $\mathsf{z}_{\mathcal{D}}^{\mathcal{P}}$ of $\mathcal{P}$ is $\mathsf{z}_{\mathcal{D}}^{\mathcal{P}} \geq \sigma_T$) using the $\mathsf{processPattern}(\mathcal{P})$ procedure. This procedure first computes the interval $C(\mathcal{P})$ (lines 7–9), and then computes the upper bound $\hat{p}_{\mathcal{P}}$ to the p-value (lines 10–12). If $C(\mathcal{P}) = \emptyset$, $\hat{p}_{\mathcal{P}}$ is set to 0 (line 10); otherwise SPuMaNTE computes $\hat{p}_{\mathcal{P}}$ using the bound from Theorem 4.3.2. SPuMaNTE uses the upper bound $\hat{p}_{\mathcal{P}}$ to decide whether $\mathcal{P}$ is significant, returning $\mathcal{P}$ in output if $\hat{p}_{\mathcal{P}} < \delta$ (line 13). Then (lines 14–15), the current pattern $\mathcal{P}$ is "grown" generating the set of closed patterns that are extension of $\mathcal{P}$ using $\mathsf{children}(\mathcal{P})$, enumerating the space of the testable patterns exhaustively in a depth-first order.

We can show the following property of SPuMaNTE.

Theorem 4.3.8. *The output of SPuMaNTE has FWER at most* α.

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 Section 4.3.2, let $\bar{E}_\varepsilon$ 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 $\Pr(\bar{E}_\varepsilon) \leq \gamma$. By employing the LAMP strategy with parameter $\alpha - \gamma$ and using the upper bound $\hat{p}_\mathcal{P}$ to decide if $\mathcal{P}$ is significant, it holds that $\Pr(F \mid E_\varepsilon) \leq \alpha - \gamma$. Therefore $\Pr(F) \leq \alpha - \gamma + \gamma = \alpha$. □

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 Theorem 4.3.2 does not always provide a tight approximation to the p-value $p_\mathcal{P}$ of a pattern $\mathcal{P}$, resulting in a potential reduction in power, even if as we show in Section 4.4.1 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 $\check{p}_\mathcal{P}$ of Section 4.3.1 in place of the upper bound of Theorem 4.3.2 in lines 11–12. While in this case there is no guarantee on the $FWER$ of the reported patterns, we show in Section 4.4.1 that $\check{p}_\mathcal{P}$ is very close to the actual p-value $p_\mathcal{P}$, leading to a relatively low risk of reporting false discoveries.

4.4 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 $\check{p}_\mathcal{P}$ from (4.3) w.r.t. the exact p-value $p_\mathcal{P}$.

- evaluate the computational performance 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 $\hat{p}_\mathcal{P}$ and of the AMIRA confidence intervals on reporting significant patterns.

Algorithm 3: SPuMANTE

Input: Dataset $\mathcal{D}$, bound $\alpha \in (0,1)$ to $FWER$, confidence par. $\gamma \in (0, \alpha)$.
Output: Set of significant patterns with $FWER \leq \alpha$.

1 $\varepsilon \leftarrow$ AMIRA $(\mathcal{D}, \gamma)$
2 $\delta \leftarrow \mathsf{correctedSignificanceThreshold}(\alpha - \gamma)$
3 $\sigma_T \leftarrow \min\{x : \check{\psi}(x) \leq \delta, 1 \leq x \leq n\}$
4 **forall** $\mathcal{P} \in$ *children*$(\emptyset)$ **do**
5 **if** $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}} \geq \sigma_T$ **then** $\mathsf{processPattern}(\mathcal{P})$
6 **Function** *processPattern*$(\mathcal{P})$
7 $C_0(\mathcal{P}) \leftarrow [\mathsf{f}_0(\mathcal{P}) - \varepsilon\frac{n}{n_0}, \mathsf{f}_0(\mathcal{P}) + \varepsilon\frac{n}{n_0}]$
8 $C_1(\mathcal{P}) \leftarrow [\mathsf{f}_1(\mathcal{P}) - \varepsilon\frac{n}{n_1}, \mathsf{f}_1(\mathcal{P}) + \varepsilon\frac{n}{n_1}]$
9 $C_{\mathcal{P}} \leftarrow C_0(\mathcal{P}) \cap C_1(\mathcal{P})$
10 **if** $C_{\mathcal{P}} = \emptyset$ **then** $\hat{p}_{\mathcal{P}} \leftarrow 0$
11 **else**
12 $\hat{p}_{\mathcal{P}} \leftarrow \mathsf{b}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1} \mid \mathsf{f}(\mathcal{P})\right)(n_0+1)(n_1+1)$
13 **if** $\hat{p}_{\mathcal{P}} \leq \delta$ **then output** $\mathcal{P}$
14 **forall** $\mathcal{P}' \in$ *children*$(\mathcal{P})$ **do**
15 **if** $\mathsf{z}^{\mathcal{P}'}_{\mathcal{D}} \geq \sigma_T$ **then** $\mathsf{processPattern}(\mathcal{P}')$

Implementation and environment We implemented SPuMANTE[1] and UT by modifying a `C` implementation of LAMP[2]. For computing the incomplete beta function in lines 9 and 12 of Algorithm 2, we use a publicly available implementation[3] based on Lentz's algorithm (Lentz, 1976). All the code was compiled with GCC 8 and run on a machine with a 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, gathered from FIMI'04[4] and libSVM[5]. Due to space constraints we only report results for three datasets (the results for other datasets are analogous). Descriptive statistics and preprocessing for these datasets are in Section 4.5.

[1]The code of SPuMANTE and the scripts to replicate all experiments are available at `https://github.com/VandinLab/SPuManTE`. See also Section 4.5.
[2]`https://github.com/fllinares/wylight`
[3]`https://github.com/codeplea/incbeta`
[4]`http://fimi.ua.ac.be`
[5]`https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

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 $[10^3, 10^6]$.

We compare SPuMaNTE to three different variants: the first, that we denote SPuMaNTE*, uses the lower bound $\check{p}_{\mathcal{P}}$ instead of the upper bound $\hat{p}_{\mathcal{P}}$ to flag significant patterns, providing increased power at the expense of relaxed guarantees on $FWER$; the second version, SPuMaNTE$^{\mathrm{C}}$, flags an itemset $\mathcal{P}$ as significant only if its confidence interval $C(\mathcal{P})$ is $C(\mathcal{P}) = \emptyset$; the last one, SPuMaNTE$^{\mathrm{n}}$, uses a naïve implementation of Algorithm 2, that enumerates all the contingency tables for every pattern $\mathcal{P}$, fixing π to $\mathsf{f}(\mathcal{P})$. However, we do not include the results for SPuMaNTE$^{\mathrm{n}}$, since its naïve enumeration strategy results in impractical running times: for $s = 10^3$, the running time of SPuMaNTE$^{\mathrm{n}}$ is always at least one order of magnitude higher than all other approaches, and could not complete in one day for $s \geq 10^4$. SPuMaNTE$^{\mathrm{n}}$ would require even more time if an expensive search over the values of π is performed to compute $p_{\mathcal{P}}$ exactly.

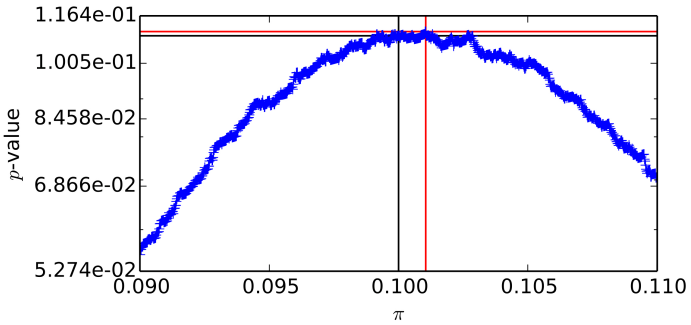


Figure 4-2: Values of $\phi(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}, \pi)$ for 10^3 equally spaced values of $\pi \in I_{\mathsf{f}(\mathcal{P})} = [0.9 \cdot \mathsf{f}(\mathcal{P}), 1.1 \cdot \mathsf{f}(\mathcal{P})]$ with $n = 10^3$, $n_1 = 500$, $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}} = 100$, $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1} = 40$. The red vertical line corresponds to $\pi^* = \arg\max_{\pi \in I_{\mathsf{f}(\mathcal{P})}} \{\phi(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}, \pi)\}$, the black vertical line to $\mathsf{f}(\mathcal{P})$, the red horizontal line to $\phi(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}, \pi^*)$, the black horizontal line to $\phi(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}, \mathsf{f}(\mathcal{P}))$.

4.4.1 Results

Quality of the lower bound Our first experiment aims at evaluating the quality of the lower bound $\check{p}_{\mathcal{P}}$. For fixed n, n_1 and $\mathcal{P}$ (described in Figure 4-2), we compute $\phi(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}, \pi)$ varying π over 10^3 equally spaced values in a fixed interval. The value $\phi(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}, \mathsf{f}(\mathcal{P}))$ coincides with our lower bound $\check{p}_{\mathcal{P}}$. Figure 4-2 shows the resulting curve: even if $\check{p}_{\mathcal{P}} \neq p_{\mathcal{P}}$ (therefore $\pi = \mathsf{f}(\mathcal{P})$ does not maximize $\phi(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}, \pi)$), using $\check{p}_{\mathcal{P}}$ provides a principled choice to obtain a very tight lower bound to $p_{\mathcal{P}}$. Similar results, not shown for space constraints, hold for different choices of n, n_1, and $\mathcal{P}$.

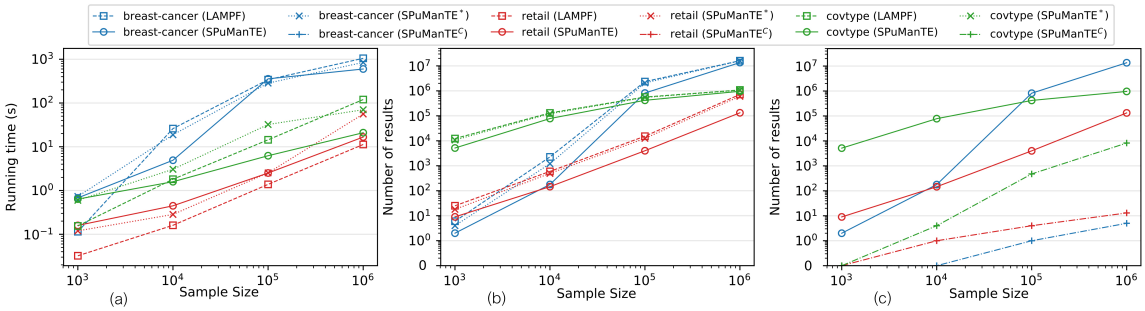


Figure 4-3: Comparison between LAMPF, SPuManTE and SPuManTE* in terms of: (a) running time; (b) number of patterns in output. In (c) we compare the number of patterns in output found with SPuManTE and SPuManTE$^{\text{C}}$.

Running time of SPuManTE In Figure 4-3.(a), we compare the running times of SPuManTE and SPuManTE* w.r.t. the state-of-the-art by LAMP with Fisher's exact test, denoted with LAMPF. Contrary to the common belief that unconditional tests are computationally expensive, SPuManTE is, in almost all cases, faster than LAMPF. These results stress the efficiency of the upper bound from Theorem 4.3.2. The only cases where SPuManTE is slower is for small sample sizes ($s \leq 10^4$), for running times ≤ 10 seconds, and for `retail` dataset. When s is small, the time to compute ε in SPuManTE dominates on the total execution time. For larger sample sizes, SPuManTE is faster than LAMPF by up to almost one order of magnitude. SPuManTE*, despite computing $\check{p}_{\mathcal{P}}$ for every $\mathcal{P}$, generally requires comparable running time w.r.t. LAMPF, thanks to our efficient strategy for computing $\check{p}_{\mathcal{P}}$ (Section 4.3.1). These results show that 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 Theorem 4.3.2 in reporting significant patterns. Figure 4-3.(b) displays the number of patterns in the output of SPuManTE, SPuManTE*, and LAMPF. The set of results reported by SPuManTE* is always a super-set of the set of results with guarantees on the $FWER$, since SPuManTE* uses a lower bound to the exact p-value. In all cases SPuManTE reports a large set of results, comparable in size with the output of SPuManTE*, therefore UT retains most of the statistical power that would be achieved with the expensive computing of the exact value of $p_{\mathcal{P}}$. We can observe that, in almost all cases, LAMPF reports more (in some cases, twice as many) patterns than SPuManTE*. This difference between unconditional and conditional procedures is due to the difference between their small p-values (see Figure 4-1), and also because the set of selected *testable* patterns by the two procedures is often different; elucidating the implication of these results is an important investigation for future research.

Lastly, we investigate the impact of using the confidence intervals in SPuMANTE. Figure 4-3.(c) shows the number of output patterns by SPuMANTE and the ones reported by SPuMANTE$^{\text{C}}$ (the variant of SPuMANTE that only checks whether $C(\mathcal{P}) = \emptyset$ to flag a pattern $\mathcal{P}$ as significant): the larger the sample, the higher is the number of patterns flagged as significant by SPuMANTE$^{\text{C}}$, 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 $C(\mathcal{P}) = \emptyset$, proving that the use of confidence intervals is a crucial component of SPuMANTE.

4.5 Proofs and Reproducibility

In this Section we present the proofs not included in the main text and describe how to reproduce our experimental results.

Missing proofs

The proofs of our results are provided here.

Theorem 4.5.1 (Theorem 4.3.2 in the main text)**.** *It holds*

$$p_{\mathcal{P}} \leq \mathsf{b}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1} \mid \mathsf{f}(\mathcal{P})\right)(n_0+1)(n_1+1) \ .$$

Proof. It holds

$$\begin{aligned}
p_{\mathcal{P}} &= \max_{\pi \in C_{\mathcal{P}}} \left\{ \sum_{(x,a) \in T(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}, \pi)} \mathsf{b}(x, a \mid \pi) \right\} \\
&\leq \max_{\pi \in C_{\mathcal{P}}} \left\{ \mathsf{b}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1} \mid \pi\right) |T(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}, \pi)| \right\} \\
&\leq \max_{\pi \in C_{\mathcal{P}}} \left\{ \mathsf{b}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1} \mid \pi\right) \right\} \max_{\pi \in C_{\mathcal{P}}} \left\{ |T(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}, \pi)| \right\} \\
&\leq \mathsf{b}\left(\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}, \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1} \mid \mathsf{f}(\mathcal{P})\right)(n_0+1)(n_1+1).
\end{aligned}$$

where in the last step we use Theorem 4.3.4 and the fact that the total number of contingency tables is $(n_0+1)(n_1+1)$, that is an upper bound to the size of $T(a, b, \pi)$ for any a, b, π. □

Proposition 4.5.2 (Theorem 4.3.4 in the main text)**.** *It holds*

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

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

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

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

Theorem 4.5.3 (Theorem 4.3.5 in the main text)**.** *Let* $C_0(\mathcal{P}) \cap C_1(\mathcal{P}) = C_{\mathcal{P}} \neq \emptyset$. *Then* $\mathsf{f}(\mathcal{P}) \in C_{\mathcal{P}}$.

Proof. We prove the result assuming $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0}/n_0 > \mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}/n_1$ (the proof for the other case is analogous) and assuming that the confidence intervals have the form provided by AMIRA. Recall that $\mathsf{f}(\mathcal{P}) = \mathsf{z}^{\mathcal{P}}_{\mathcal{D}}/n$. $C_0(\mathcal{P}) \cap C_1(\mathcal{P}) \neq \emptyset$ is equivalent to

$$\frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}}{n_1} + \varepsilon\frac{n}{n_1} \geq \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0}}{n_0} - \varepsilon\frac{n}{n_0}. \tag{4.5}$$

and that proving $\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}/n \in C_0(\mathcal{P}) \cap C_1(\mathcal{P})$ corresponds to prove that

$$\frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}}{n_1} + \varepsilon\frac{n}{n_1} \geq \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}}{n} \tag{4.6}$$

and

$$\frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0}}{n_0} - \varepsilon\frac{n}{n_0} \leq \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}}{n} \tag{4.7}$$

both hold. Equation (4.5) above is equivalent to

$$\frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0}}{n_0} - \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}}{n_1} \leq \varepsilon n\left(\frac{1}{n_0} + \frac{1}{n_1}\right) \ . \tag{4.8}$$

It holds

$$\frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}}{n} = \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}}{n_1} + \frac{n_0}{n}\left(\frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0}}{n_0} - \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}}{n_1}\right)$$

and from (4.8) we derive

$$\begin{aligned}\frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}}{n} &= \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}}{n_1} + \frac{n_0}{n}\left(\frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0}}{n_0} - \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}}{n_1}\right) \leq \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}}{n_1} + \frac{n_0}{n}\varepsilon n\left(\frac{1}{n_0} + \frac{1}{n_1}\right) \\ &= \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}}{n_1} + n_0\varepsilon\left(\frac{1}{n_0} + \frac{1}{n_1}\right) = \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}}{n_1} + \varepsilon\left(1 + \frac{n_0}{n_1}\right) = \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}}{n_1} + \varepsilon\frac{n}{n_1},\end{aligned}$$

that proves (4.6).

For (4.7), it holds

$$\frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}}{n} = \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0}}{n_0} - \frac{n_1}{n}\left(\frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0}}{n_0} - \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}}{n_1}\right)$$

and from (4.8) we derive

$$\begin{aligned}\frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}}}{n} &= \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0}}{n_0} - \frac{n_1}{n}\left(\frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0}}{n_0} - \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^1}}{n_1}\right) \geq \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0}}{n_0} - \frac{n_1}{n}\varepsilon n\left(\frac{1}{n_0} + \frac{1}{n_1}\right) \\ &= \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0}}{n_0} - n_1\varepsilon\left(\frac{1}{n_0} + \frac{1}{n_1}\right) = \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0}}{n_0} - \varepsilon\left(1 + \frac{n_1}{n_0}\right) = \frac{\mathsf{z}^{\mathcal{P}}_{\mathcal{D}^0}}{n_0} - \varepsilon\frac{n}{n_0},\end{aligned}$$

that proves (4.7). □

Theorem 4.5.4 (Theorem 4.3.6 in the main text)**.**

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

Proof. We assume $n_1 \leq n_0$. Let $\check{a} = \check{\mathsf{z}}_{1,x}$ and $\hat{a} = \hat{\mathsf{z}}_{1,x}$. We first prove that

$$\min_a\{\mathsf{b}(x, a, \pi)\} = \mathsf{b}(x, \hat{a}, \pi) \tag{4.9}$$

by observing that, $\forall a \in [\check{a}, \hat{a}]$,

$$\frac{\mathsf{b}(x, a, \pi)}{\mathsf{b}(x, \hat{a}, \pi)} = \frac{B(x - a, n - n_1, \pi)B(a, n_1, \pi)}{B(x - \hat{a}, n - n_1, \pi)B(\hat{a}, n_1, \pi)} = \frac{\binom{n-n_1}{x-a}\binom{n_1}{a}}{\binom{n-n_1}{x-\hat{a}}\binom{n_1}{\hat{a}}} \geq 1.$$

A direct consequence of (4.9) and the definition of $T(x, a, \pi)$ is

$$T(x, \hat{a}, \pi) \subseteq T(x, a, \pi), \forall a \in [\check{a}, \hat{a}]. \tag{4.10}$$

Then (4.10) leads to

$$\frac{\phi(x, a, \pi)}{\phi(x, \hat{a}, \pi)} = \frac{\sum_{(y,b)\in T(x,a,\pi)} \mathsf{b}(y, b \mid \pi)}{\sum_{(y,b)\in T(x,\hat{a},\pi)} \mathsf{b}(y, b \mid \pi)} \geq 1, \forall a \in [\check{a}, \hat{a}],$$

that proves the statement. □

Theorem 4.5.5 (Theorem 4.3.7 in the main text)**.** *Let* $y \in [0, n]$, $b \in [\check{\mathsf{z}}_{1,y}, \hat{\mathsf{z}}_{1,y}]$, *and* $\pi \in (0, 1)$. *Let*

$$A_1 = \{a_1 : \mathsf{b}\left(a_1 + \lfloor(n_0 + 1)\pi\rceil, a_1 \mid \pi\right) > \mathsf{b}(y, b \mid \pi)\},$$

and define the set $A_{0,a_1} = \{a_0 : \mathsf{b}\left(a_1 + a_0, a_1 \mid \pi\right) \leq \mathsf{b}(y, b \mid \pi)\}$. *Then*

$$\sum_{(x,a)\in T(y,b,\pi)} \mathsf{b}(x, a \mid \pi) = \sum_{a_1 \notin A_1} B(a_1, n_1, \pi) + \sum_{a_1 \in A_1} \left(B(a_1, n_1, \pi) \sum_{a_0 \in A_{0,a_1}} B(a_0, n_0, \pi) \right),$$

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 $\sum_{(x,a)\in T(y,b,\pi)} \mathsf{b}(x, a \mid \pi)$ as

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

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

$$\mathsf{b}\left(a_1 + \lfloor (n_0+1)\pi \rceil, a_1 \mid \pi\right) \geq \mathsf{b}\left(a_1 + a_0, a_1 \mid \pi\right), \forall a_0 \in [0, n_0] \ .$$

Given the above and the definition of A_1 and A_{0,a_1}, we obtain

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

□

Proposition 4.5.6. *The time complexity of Algorithm 2 is*

$$\mathcal{O}\left(\log(n_0) + n + La\right) \ ,$$

where $\mathcal{O}\left(La\right)$ *is the time complexity of Lentz's algorithm.*

Proof. The $\mathcal{O}(n)$ term follows from the following observations: all quantities to be required inside the **forall** loop can be be computed "from scratch" at the first iteration, and then updated with few operations at each new iteration. Using this strategy, it is simple to show that the overall cumulative work of the **forall** loop over all its $|A_1| \leq n_1$ iterations does not exceed $\mathcal{O}(n)$. This is because the indexes a' and a''

are modified from their values computed at the first iterations at most n_0 times, over all iterations. Therefore, the other terms are referred to the work done at the first iteration of the loop; operations outside the loop are $\mathcal{O}(n_1)$ for the computation of A_1, therefore included in the $\mathcal{O}(n)$ term. The $\mathcal{O}(\log(n_0))$ term follows from a binary search over n_0 sorted elements, that is performed on the first iteration of the **forall** loop. The $\mathcal{O}(La)$ term follows since the total number of calls to Lentz's algorithm is constant, as tails of binomial distribution are updated during the iterations. □

Reproducibility

We now describe how to reproduce our experimental results. Code and data are available at `https://github.com/VandinLab/SPuManTE`.

Datasets and preprocessing The statistics of the datasets we analysed are described in Table 4.1. 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.

dataset	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{I}\rvert$	avg	n_1/n
breast cancer	12,773	1,129	6.7	0.09
retail(U)	88,162	16,470	10.3	0.47
covtype	581,012	64	11.9	0.49

Table 4.1: Datasets statistics. For each dataset we report: name (see Section 4.4 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 $\mathcal{D}^1$.

Reproducing our simulations The plot of Figure 4-1 can be created with the Python script `fisher_simulations.py` in the `scripts/` folder. The results of Figure 4-2 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 ad LAMPF in parallel, and can be used to reproduce all the experiments described in Section 4.4.

Chapter 5

Sampling-based Methods for Frequent k-mers Approximations

5.1 Introduction

The analysis of substrings of length k, called *k-mers*, is ubiquitous in biological sequence analysis and is among the first steps of processing pipelines for a wide spectrum of applications, including: de novo assembly (Pevzner et al., 2001; Zerbino and Birney, 2008), error correction (Kelley et al., 2010; Salmela et al., 2016), repeat detection (Li and Waterman, 2003), genome comparison (Sims et al., 2009), digital normalization (Brown et al., 2012), RNA-seq quantification (Patro et al., 2014; Zhang and Wang, 2014), metagenomic reads classification (Wood and Salzberg, 2014) and binning (Girotto et al., 2016), fast search-by-sequence over large high-throughput sequencing repositories (Solomon and Kingsford, 2016). A fundamental task in k-mer analysis is to compute the frequency of all k-mers, with the goal to distinguish frequent k-mers from infrequent k-mers (Marçais and Kingsford, 2011; Melsted and Pritchard, 2011). For example, this task is relevant in the analysis of high-throughput sequencing data, since infrequent k-mers are often assumed to result from sequencing errors. For several applications, the computation of k-mer frequencies is among the most computationally demanding steps of the analysis.

Many algorithms have been proposed for computing the exact frequency of all k-mers, such as Tallymer (Kurtz et al., 2008), Jellyfish (Marçais and Kingsford, 2011), BFCounter (Melsted and Pritchard, 2011), DSK (Rizk et al., 2013), KAnalyze (Audano and Vannberg, 2014), Turtle (Roy et al., 2014), KMC 3 (Kokot et al., 2017), and Squeakr-exact (Pandey et al., 2017). These methods typically perform a linear scan of the sequences to analyze, and use a combination of parallelism and efficient data structures (such as Bloom filters and Hash tables) to maintain membership and counting information associated to all k-mers. Since the computation of exact k-mer frequencies is computationally demanding, in particular for large sequence analysis or for high-throughput sequence datasets, recent methods have focused on providing approximate solution to the problem, improving the time and memory requirements. KmerStream (Melsted and Halldórsson, 2014), Kmerlight (Sivadasan et al., 2016) and ntCard (Mohamadi et al., 2017) proposed streaming approaches for the approximation of the k-mer frequencies histogram. KmerGenie (Chikhi and Medvedev, 2013) performs a linear scan of the input, counting a random subset (chosen before processing the dataset) of all possible k-mers to approximate the abundance histogram, providing an exploratory tool to choose the value of k. khmer (Zhang et al., 2014) and the recently proposed Squeakr (Pandey et al., 2017) rely on probabilistic data structures to approximate the counts of individual k-mers. With the only exception of KmerGenie, all these methods processes *all* the k-mer occurrences in the input dataset; in addition, all the aforementioned approximate methods that report the counts of individual k-mers do not provide simultaneous estimates with rigorous guarantees for all the counts k-mers that are provided in output.

All the methods cited above try to estimate the frequency of *all* k-mers or of all k-mers that appear at least few times (e.g., twice) in the dataset. While this is crucial

in some applications (e.g., in genome assembly k-mers that occur exactly once often represents sequencing errors and it is therefore important to estimate the count of all observed k-mers), in other applications this is less justified. For example, in the comparison of high-throughput sequencing metagenomic datasets, *abundance-based distances or dissimilarities* (e.g., the Bray-Curtis dissimilarity) between k-mer counts of two datasets are often used (Benoit et al., 2016; Danovaro et al., 2017; Dickson et al., 2017) to assess the distance between the corresponding datasets. In contrast to *presence-based distances* (Ondov et al., 2016) (e.g., Jaccard distance), abundance-based distances take into account the frequency of each k-mer, with frequent k-mers contributing more to the distance than k-mers that appear with low frequency, but still more than a handful of times, in the dataset. Thus, two natural questions are (i) whether the results obtained considering all k-mers can be estimated by considering the abundances of frequent k-mers only, and (ii) if the abundances of frequent k-mers can be computed more efficiently than the counts of all k-mers. Recently, preliminary work (Hrytsenko et al., 2018) has shown that, for the cosine distance and $k = 12$, the answer to the first question is positive, and in Section 5.4 we show that this indeed the case for larger values of k and other abundance-based distances as well as presence-based distances (e.g., the Jaccard distance). To the best of our knowledge, the second question is hitherto unexplored. In addition, considering only frequent k-mers allows to focus on the most reliable information in a metagenomic dataset, since a high stochastic variability in low frequency k-mers is to be expected due to the sampling process inherent in sequencing.

A natural approach to reduce time and memory requirements for frequency estimation problems is to process only a portion of the data, for example by *sampling* some parts of a dataset. Sampling approaches are appealing because infrequent k-mers naturally tend to appear with lower probability in a sample, allowing to directly focus on frequent k-mers in subsequent steps. However, major challenges in sampling approaches are (i) to provide rigorous guarantees relating the results obtained by processing the sample and the results that would be obtained from the whole dataset, and (ii) to provide effective bounds on the size of the sample required to achieve such guarantees. The application of sampling to k-mers is even more challenging than in other scenarios since, for values of k in the typical range of interest to applications (e.g., 20-60), even the most frequent k-mers have relatively low frequency in the data. To the best of our knowledge, no approach based on sampling a portion of the input dataset has been proposed to approximate frequent k-mers and their frequencies while providing rigorous guarantees.

Our Contribution. We study the problem of approximating frequent k-mers, i.e., k-mers that appear with frequency above a user-defined threshold θ in a high-throughput sequencing dataset. In these regards, our contributions are fourfold. First, we define a rigorous definition of approximation, governed by an accuracy parameter ε. Second, we propose a new method, Sampling Algorithm for K-mErs approxIMAtion (`SAKEIMA`), to obtain an approximation to the set of frequent k-mers using *sampling*.

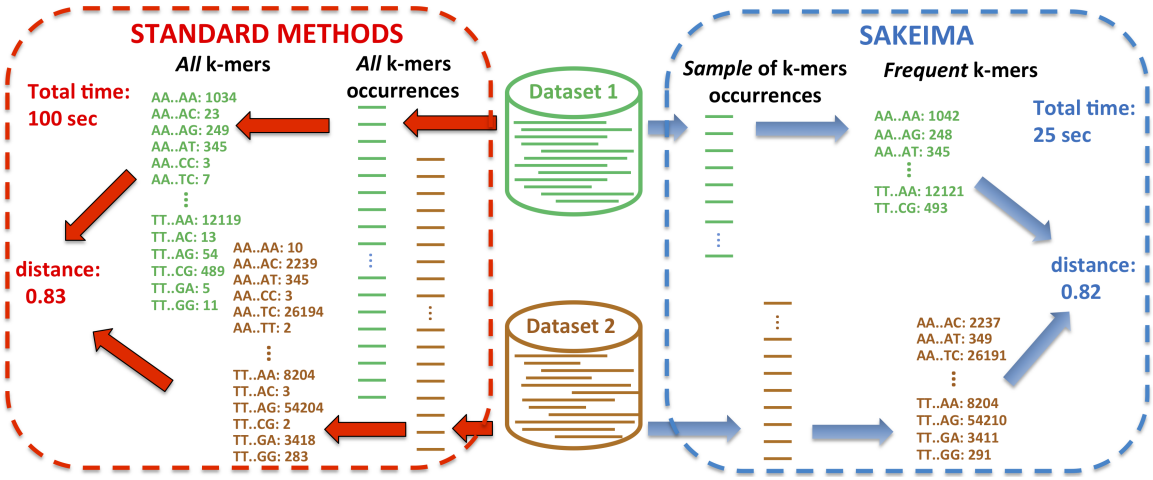


Figure 5-1: SAKEIMA computes a fast and rigorous approximation of the *frequent* k-mers in a high-throughput sequencing dataset by sampling a fraction of all k-mer occurrences in a dataset, providing a significant speed-up for the computation of k-mer's abundance-based distances between datasets of reads (e.g., in metagenomics).

SAKEIMA is based on a sampling scheme that goes beyond naïve sampling of k-mers and allows to estimate k-mers of relatively low frequency considering only a fraction of all k-mers occurrences in the dataset. Third, we provide analytical bounds to the sample size needed to obtain rigorous guarantees on the accuracy of the estimated k-mer frequencies, with respect to the ones measured on the entire dataset. Our bounds are based on the notion of Vapnik-Chervonenkis (VC) dimension, a fundamental concept from Statistical Learning Theory, which has been used to design efficient algorithms to identify frequent patterns in other scenarios (Riondato and Upfal, 2015; Riondato and Kornaropoulos, 2016; Servan-Schreiber et al., 2018b). To our knowledge, ours is the first method that applies concepts from *statistical learning* to provide a rigorous approximation of the k-mer frequencies. Fourth, we use SAKEIMA to extract frequent k-mers from metagenomic datasets from the Human Microbiome Project (HMP) and to approximate abundance-based and presence-based distances among such datasets, showing that SAKEIMA allows to accurately estimate such distances by analyzing only a fraction of the entire dataset, resulting in a significant speed-up.

Our approach is essentially orthogonal to previous work: most of the exact or approximate algorithms for k-mer counting can be applied to the sample extracted by SAKEIMA, that can therefore be used *before* applying previously proposed methods, thus reducing their computational requirements while providing rigorous guarantees on the results w.r.t. to the entire dataset. While we present our methodology in the case of finding frequent k-mers from a set of sequences representing a high-throughput sequencing dataset of short reads, our results can be applied to datasets of long reads and to whole-genome sequences as well.

5.2 Preliminaries

Let a dataset $\mathcal{D}$ be a bag of n reads $\mathcal{D} = \{r_0, \ldots, r_{n-1}\}$, where each read r_i, $0 \leq i \leq n-1$, is a string of length n_i from an alphabet Σ of cardinality $|\Sigma| = \sigma$. For $j \in \{0, \ldots, n_i - 1\}$, let $r_i[j]$ be the j-th character of r_i. For a given integer $k \leq \min_i\{n_i : r_i \in \mathcal{D}\}$, we define a k-mer A as a string of length k from Σ, that is $A \in \Sigma^k$. We say that a k-mer A *appears* in r_i at position $j \in \{0, \ldots, n_i - k\}$ if $r_i[j+h] = A[h], \forall h \in \{0, \ldots, k-1\}$. For every $i, 0 \leq i \leq n-1$, and every $j \in \{0, \ldots, n_i - k\}$, we define the indicator function $\phi_{r_i,A}(j)$ that is 1 if the k-mer A appears in r_i at position j, while $\phi_{r_i,A}(j) = 0$ otherwise. The total number of k-mers in $\mathcal{D}$ is $t_{\mathcal{D},k} = \sum_{i=0}^{n-1}(n_i - k + 1)$. We define the *support* $o_{\mathcal{D}}(A)$ of a k-mer A as the number of distinct positions in $\mathcal{D}$ where A appears: $o_{\mathcal{D}}(A) = \sum_{i=0}^{n-1}\sum_{j=0}^{n_i-k}\phi_{r_i,A}(j)$. We define the *frequency* $f_{\mathcal{D}}(A)$ of A in $\mathcal{D}$ as the ratio between the number of distinct positions where A appears in $\mathcal{D}$ and the total number of k-mers in $\mathcal{D}$: $f_{\mathcal{D}}(A) = o_{\mathcal{D}}(A)/t_{\mathcal{D},k}$.

5.2.1 Frequent k-mers and Approximations

We are interested in obtaining the set $FK(\mathcal{D}, k, \theta)$ of frequent k-mers in a dataset $\mathcal{D}$ with respect to a minimum frequency threshold θ, defined as follows.

Definition 5.2.1. Given a dataset $\mathcal{D}$, an integer $k > 0$, and a frequency threshold $\theta \in (0, 1]$, the set $FK(\mathcal{D}, k, \theta)$ of *Frequent k-Mers in $\mathcal{D}$ w.r.t. θ* is the collection of all k-mers with frequency at least θ in $\mathcal{D}$ and of their corresponding frequencies in $\mathcal{D}$:

$$FK(\mathcal{D}, k, \theta) = \{(A, f_{\mathcal{D}}(A)) : f_{\mathcal{D}}(A) \geq \theta\}. \tag{5.1}$$

$FK(\mathcal{D}, k, \theta)$ can be computed with a single scan of all the k-mers occurrences in $\mathcal{D}$ maintaining the k-mers supports in an appropriate data structure; however, when $\mathcal{D}$ is extremely large and k is not small, the exact computation of $FK(\mathcal{D}, k, \theta)$ is extremely demanding in terms of time and memory, since the number of k-mers grows exponentially with k. In this case, a fast to compute *approximation* of the set $FK(\mathcal{D}, k, \theta)$ may be preferable, provided it ensures rigorous guarantees on its quality. In this work, we focus on the following approximation.

Definition 5.2.2. Given a dataset $\mathcal{D}$, an integer $k > 0$, a frequency threshold $\theta \in (0, 1]$, and a constant $\varepsilon \in (0, \theta)$, an *$\varepsilon$-approximation* of $FK(\mathcal{D}, k, \theta)$ is a collection $C = \{(A, f_A) : f_A \in (0, 1]\}$ such that:

- for any $(A, f_{\mathcal{D}}(A)) \in FK(\mathcal{D}, k, \theta)$ there is a pair $(A, f_A) \in C$;
- for any $(A, f_A) \in C$ it holds that $f_{\mathcal{D}}(A) \geq \theta - \varepsilon$;
- for any $(A, f_A) \in C$ it holds that $|f_{\mathcal{D}}(A) - f_A| \leq \varepsilon/2$.

The definition above guarantees that every frequent k-mer of $\mathcal{D}$ is in the approximation and that no k-mer with frequency $< \theta - \varepsilon$ is in the approximation. The third condition guarantees that the estimated frequency f_A of A in the approximation is close (i.e, within $\varepsilon/2$) to the frequency $f_{\mathcal{D}}(A)$ of A in $\mathcal{D}$. It is easy to show that obtaining a ε-approximation of $FK(\mathcal{D}, k, \theta)$ with absolute certainty requires to process all k-mers in $\mathcal{D}$.

5.2.2 Simple Sampling-Based Algorithms and Bounds

We aim to provide an approximation to $FK(\mathcal{D}, k, \theta)$ with *sampling*, by processing only *randomly selected portions* of $\mathcal{D}$. The simplest sampling scheme is the one in which a random sample is a bag P of m positions taken uniformly at random, with replacement, from the set $P_{\mathcal{D},k} = \{(i, j) : i \in [0, n-1], j \in [0, n_i - k]\}$ (note that $|P_{\mathcal{D},k}| = t_{\mathcal{D},k}$) of all positions where k-mers occurs in the dataset $\mathcal{D}$, corresponding to m occurrences of k-mers (with repetitions) taken uniformly at random. Given such sample P, an integer $k > 0$, and a minimum frequency threshold $\theta \in (0, 1]$ one can define the set of frequent k-mers (and their frequencies) in the sample P as $FK(P, k, \theta) = \{(A, f_P(A)) : f_P(A) \geq \theta\}$, where $f_P(A)$ is the frequency of k-mer A in the sample.

Obtaining a ε-approximation from a random sample with absolute certainty is impossible, thus we focus on obtaining a ε-approximation with probability $1 - \delta > 0$, where $\delta \in (0, 1)$ is a *confidence* parameter, whose value is provided by the user. Intuitively, the set $FK(\mathcal{D}, k, \theta)$ of frequent k-mers is well approximated by the set of frequent k-mers in a random sample P when P is sufficiently large. One natural question regards how many samples are needed to obtain the desired ε-approximation. By using Hoeffding's inequality (Mitzenmacher and Upfal, 2017) to bound the deviation of the frequency of a k-mer A in the sample from $f_{\mathcal{D}}(A)$ and a union bound on the maximum number σ^k of k-mers, where $\sigma = |\Sigma|$, we have the following result that provides a first such bound, and a corresponding first algorithm to obtain a ε-approximation to $FK(\mathcal{D}, k, \theta)$.

Proposition 5.2.3. *Consider a sample P of size m of $\mathcal{D}$. If $m \geq \frac{2}{\varepsilon^2}\left(\ln\left(2\sigma^k\right) + \ln\left(\frac{1}{\delta}\right)\right)$ for fixed $\varepsilon \in (0, \theta), \delta \in (0, 1)$, then, with probability $\geq 1 - \delta$, $FK(P, k, \theta - \varepsilon/2)$ is a ε-approximation of $FK(\mathcal{D}, k, \theta)$.*

Proof. We first prove that when $m \geq \frac{2}{\varepsilon^2}\left(\ln\left(2\sigma^k\right) + \ln\left(\frac{1}{\delta}\right)\right)$, then, with probability $\geq 1 - \delta$, for every k-mer A simultaneously we have $|f_P(A) - f_{\mathcal{D}}(A)| \leq \varepsilon/2$.

For an arbitrary k-mer A, given the definition of $f_P(A)$ we have that $f_P(A) = \sum_{(i,j)\in P} \phi_{r_i,A}(j)/m$ where $\sum_{(i,j)\in P} \phi_{r_i,A}(j)$ is the sum of m 0-1 independent random variables. Since $\mathbb{E}[\phi_{r_i,A}(j)] = f_{\mathcal{D}}(A)$, we have that $\mathbb{E}[f_P(A)] = f_{\mathcal{D}}(A)$, and by Ho-

effding's inequality (Mitzenmacher and Upfal, 2017) we have

$$\Pr(|f_P(A)-f_{\mathcal{D}}(A)| \geq \varepsilon) = \Pr\left(\left|\sum_{(i,j)\in P} \phi_{r_i,A}(j) - m f_{\mathcal{D}}(A)\right| \geq m\varepsilon\right) \leq 2e^{\frac{-2m^2\varepsilon^2}{m}} = 2e^{-2m\varepsilon^2}. \tag{5.2}$$

Now define the event $E_A =$ "$|f_P(A)-f_{\mathcal{D}}(A)| \leq \varepsilon/2$" and let $\bar{E}_A$ be the complementary event. From Equation 5.2 and the choice of m, $\Pr(\bar{E}_A) \leq 2e^{-m\varepsilon^2/2} = \delta/\sigma^k$. By union bound, the probability that at least one $\bar{E}_A$ holds is bounded by $\sum_{A\in\Sigma^k} \Pr(\bar{E}_A) \leq \delta$. Therefore with probability at least $1-\delta$ all events E_A hold.

We now prove that when $|f_P(A) - f_{\mathcal{D}}(A)| \leq \varepsilon/2$ for every k-mer A, then $FK(P,k,\theta-\varepsilon/2)$ is a ε-approximation of $FK(\mathcal{D},k,\theta)$. Consider an arbitrary pair $(A, f_{\mathcal{D}}(A)) \in FK(\mathcal{D},k,\theta)$. By the definition of $FK(\mathcal{D},k,\theta)$ we have that $f_{\mathcal{D}}(A) \geq \theta$, and, since $|f_P(A) - f_{\mathcal{D}}(A)| \leq \varepsilon/2$, we have that $f_P(A) \geq \theta - \varepsilon/2$, that is there is a pair $(A, f_A) \in FK(P,k,\theta-\varepsilon/2)$. Now consider a k-mer A with $f_{\mathcal{D}}(A) < \theta - \varepsilon$: since $|f_P(A) - f_{\mathcal{D}}(A)| \leq \varepsilon/2$ we have that $f_P(A) \leq f_{\mathcal{D}}(A) + \varepsilon/2 < \theta - \varepsilon/2$, that is there is no pair $(A, f_A) \in FK(P,k,\theta-\varepsilon/2)$. □

In addition, by using known results in Statistical Learning Theory (Vapnik and Chervonenkis, 1971; Mitzenmacher and Upfal, 2017) relating the VC dimension (see Section 5.3 for its definition) of a family of functions to a newly derived bound on the family of functions $\{f_{\mathcal{D}}(A)\}$, we obtain the following improved bound and algorithm. (The derivation is in Section 5.5.)

Proposition 5.2.4. *Let P be a sample of size m of $\mathcal{D}$. For fixed $\varepsilon \in (0,\theta), \delta \in (0,1)$, if $m \geq \frac{2}{\varepsilon^2}\left(1 + \ln\left(\frac{1}{\delta}\right)\right)$ then $FK(P,k,\theta-\varepsilon/2)$ is an ε-approximation for $FK(\mathcal{D},k,\theta)$ with probability $\geq 1-\delta$.*

5.3 Advanced and Practical Bounds and Algorithms for k-mer Approximations

While the bound of Proposition 5.2.4 significantly improves the simple bounds of Section 5.2.3, since the factor $\ln(2\sigma^k)$ has been reduced to 1, it still has an inverse quadratic dependency with respect to the accuracy parameter ε, that is problematic when the quantities to estimate are small. In these cases, one needs a small ε to produce a meaningful approximation (since $\varepsilon < \theta$), and the inverse quadratic dependence of the sample size from ε often results in a sample size larger than the entire input, defeating the purpose of sampling. The case of k-mers is particularly challenging, since the sum $\sum_{A\in\Sigma^k} f_{\mathcal{D}}(A)$ of all k-mer frequencies is exactly 1. Therefore the higher the number of distinct k-mers appearing in the input, the lower their frequencies will be, with the consequence that θ (and therefore ε) typically needs to be set to a very low value. For example, a typical dataset from the Human Microbiome Project

(HMP) has $n \approx 10^8$ reads of (average) length ≈ 100: therefore if we are interested in k-mers for $k = 31$, by setting $\delta = 0.05$ the bound of Section 5.2.2 gives $\varepsilon \approx 10^{-5}$, that is only k-mers with frequency $\geq 10^{-5}$ could be reliably reported by sampling. However, in datasets we considered, no or a very small number (≤ 30) of k-mers have frequency $\geq 10^{-5}$, therefore according to the result from Section 5.2.2 we cannot obtain a meaningful approximation of k-mers and their frequencies (see also Section 5.5 for more details). In the remainder of this section we develop more refined sampling schemes and estimation techniques leading to a practical sampling-based algorithm.

5.3.1 Sampling Bags of Positions and VC dimension Bound

We propose a method to provide an efficiently computable approximation to $FK(\mathcal{D}, k, \theta)$ when the minimum frequency θ is low, by properly defining samples so that any k-mer A will appear in a sample with probability higher than $f_{\mathcal{D}}(A)$, thus lessening the dependence of the sample size from $1/\varepsilon^2$. For this to be achievable, we need to relax the notion of approximation defined in Section 5.2. In particular, the guarantees, provided by our method, in such relaxed approximation are that *all* k-mers with frequency above θ', with θ' slightly higher than θ, are reported in output, and that no k-mer having frequency below $\theta - \varepsilon$ is reported in output. (See Proposition 5.3.5 for the definition of θ'.) Our experiments show that the fraction of k-mers having frequency $\in [\theta, \theta')$ which are non reported is very small. Our method works by sampling *bags of positions* instead of single positions. In particular, an element of the sample is now a set of ℓ positions chosen independently at random from the set $P_{\mathcal{D},k}$ of all positions.

Let $I_\ell = \{(i_1, j_1), (i_2, j_2), \ldots, (i_\ell, j_\ell)\}$ be a bag of ℓ positions for k-mers in $\mathcal{D}$, chosen uniformly at random from the set $P_{\mathcal{D},k}$. We define the indicator functions $\hat{\phi}_A(I_\ell)$ that, for a given bag I_ℓ of ℓ positions, is equal to 1 if k-mer A appears in *at least* one of the ℓ positions in I_ℓ and is equal to 0 otherwise. That is $\hat{\phi}_A(I_\ell) = \min\left\{1, \sum_{(i,j)\in I_\ell} \phi_{r_i,A}(j)\right\}$. We define the *$\ell$-positions sample* P_ℓ as a bag of m bags $\{I_{\ell,0}, I_{\ell,1}, \ldots, I_{\ell,m-1}\}$, where each $I_{\ell,j}, 0 \leq j \leq m-1$ is a bag of ℓ positions, sampled independently, and

$$\hat{f}_{P_\ell}(A) = \frac{1}{m} \sum_{I_{\ell,i} \in P_\ell} \frac{\hat{\phi}_A(I_{\ell,i})}{\ell}. \tag{5.3}$$

Intuitively, $\hat{f}_{P_\ell}(A)$ is the biased version of the unbiased estimator

$$f_{P_\ell}(A) = \frac{1}{m} \sum_{I_{\ell,i} \in P_\ell} \frac{\sum_{(i,j)\in I_{\ell,i}} \phi_{r_i,A}(j)}{\ell} \tag{5.4}$$

of $f_{\mathcal{D}}(A)$, where the bias arises from considering a value of 1 every time $\sum_{(i,j)\in I_{\ell,i}} \phi_{r_i,A}(j) > 1$.

In our analysis we use the VC dimension (Vapnik and Chervonenkis, 1971; Vapnik,

1998), a statistical learning concept that measures the expressivity of a family of binary functions. We define a range space Q as a pair $Q = (X, R_X)$ where X is a finite or infinite set and R_X is a finite or infinite family of subsets of X. The members of R_X are called *ranges*. Given $D \subset X$, the *projection* of R_X on D is defined as $proj_{R_X}(D) = \{r \cap D : r \in R_X\}$. We say that D is *shattered by* R_X if $proj_{R_X}(D) = 2^{|D|}$. The *VC dimension of* Q, denoted as $VC(Q)$, is the maximum cardinality of a subset of X shattered by R_X. If there are arbitrary large shattered subsets of X shattered by R_X, then $VC(Q) = \infty$.

A finite bound on the VC dimension of a range space Q implies a bound on the number of random samples required to obtain a good approximation of its ranges, defined as follows.

Definition 5.3.1. Let $Q = (X, R_X)$ be a range space and let D be a finite subset of X. For $\varepsilon \in (0, 1]$, a subset B of D is an *ε-approximation of D* if for all $r \in R_X$ we have: $\left|\frac{|D \cap r|}{|D|} - \frac{|B \cap r|}{|B|}\right| \leq \varepsilon/2$.

The following result (Mitzenmacher and Upfal, 2017) relates ε and the probability that a random sample of size m is an ε-approximation for a range space of VC dimension at most v.

Proposition 5.3.2. *There is an absolute positive constant c such that if (X, R_X) is a range-space of VC dimension at most v, D is a finite subset of X, and $0 < \varepsilon$, $\delta < 1$, then a random subset $B \subset D$ of cardinality m with $m \geq \frac{4c}{\varepsilon^2}\left(v + \ln\left(\frac{1}{\delta}\right)\right)$ is a ε-approximation of D with probability at least $1 - \delta$.*

The universal constant c has been experimentally estimated to be at most 0.5 (Löffler and Phillips, 2009).

We now prove an upper bound to the VC dimension $VC(Q)$ of the range space Q associated to the class of functions $\hat{\phi}_A$ that grows sub-linearly with respect to ℓ. To this aim, we first define the range space associated to bags of ℓ positions of k-mers.

Definition 5.3.3. Let $\mathcal{D}$ be a dataset of n reads and let k and ℓ be two integers ≥ 1. We define $Q = (X_{\mathcal{D},k,\ell}, R_{\mathcal{D},k,\ell})$ to be the following range space:

- $X_{\mathcal{D},k,\ell}$ is the set of all bags of ℓ positions of k-mers in $\mathcal{D}$, that is the set of all possible subsets, with repetitions, of size ℓ from from $P_{\mathcal{D},k}$;
- $R_{\mathcal{D},k,\ell} = \{P_{\mathcal{D},\ell}(A) | A \in \Sigma^k\}$ is the family of sets of starting positions of k-mers, such that for each k-mer A, the set $P_{\mathcal{D},\ell}(A)$ is the set of all bags of ℓ starting positions in $\mathcal{D}$ where A appears at least once.

We prove the following results on the VC dimension of the above range space.

Proposition 5.3.4. *Let Q be the range space from Definition 5.3.3. Then:*

$$VC(Q) \leq \lfloor \log_2(\ell) \rfloor + 1 \ .$$

Proof. If $VC(Q) \geq v$, then there must exists a set $Z \subseteq X_{\mathcal{D},k,\ell}$ with $|Z| = v$ that is shattered. This means that 2^v subsets of Z must be in projection of $R_{\mathcal{D},k,\ell}$ on Z. If this is true, then every element of Z needs to belong to exactly 2^{v-1} such sets. Therefore, every element of Z needs to contain at least $\ell = 2^{v-1}$ distinct k-mers. This implies that $v \leq \log_2(\ell) + 1$, and the thesis follows. □

Using the result above, we prove the following.

Proposition 5.3.5. *Let $\ell \geq 1$ be an integer and P_ℓ be a bag of m bags of ℓ positions of $\mathcal{D}$ with*

$$m \geq \frac{2}{(\ell\varepsilon)^2}\left(\lfloor \log_2 \min(2\ell, \sigma^k)\rfloor + \ln\left(\frac{1}{\delta}\right)\right). \tag{5.5}$$

Then, with probability at least $1-\delta$:

- *for any k-mer $A \in FK(\mathcal{D}, k, \theta)$ such that*

  $$f_{\mathcal{D}}(A) \geq \theta' = 1 - (1-\ell\theta)^{1/\ell} \ ,$$

  it holds $\hat{f}_{P_\ell}(A) \geq \theta - \varepsilon/2$;

- *for any k-mer A with $\hat{f}_{P_\ell}(A) \geq \theta - \varepsilon/2$ it holds $f_{\mathcal{D}}(A) \geq \theta - \varepsilon$;*

- *for any k-mer $A \in FK(\mathcal{D}, k, \theta)$ it holds $f_{\mathcal{D}}(A) \geq \hat{f}_{P_\ell}(A) - \varepsilon/2$;*

- *for any k-mer A with $\hat{f}_{P_\ell}(A) - \varepsilon/2 \geq 0$, it holds*

  $$f_{\mathcal{D}}(A) \geq 1 - (1 - \ell(\hat{f}_{P_\ell}(A) - \varepsilon/2))^{1/\ell} \ ;$$

- *for any k-mer A with $\ell(\hat{f}_{P_\ell}(A) + \varepsilon/2) \leq 1$ it holds*

  $$f_{\mathcal{D}}(A) \leq 1 - (1 - \ell(\hat{f}_{P_\ell}(A) + \varepsilon/2))^{1/\ell} \ .$$

Proof. For a given k-mer A, consider the event $E_A =$ "$|\mathbb{E}[\hat{f}_{P_\ell}(A)] - \hat{f}_{P_\ell}(A)| \leq \varepsilon/2$". Note that it is equivalent to "$|\mathbb{E}[\ell\hat{f}_{P_\ell}(A)] - \ell\hat{f}_{P_\ell}(A)| \leq \ell\varepsilon/2$" and that $\ell\hat{f}_{P_\ell}(A) = \frac{1}{m}\sum_{i=0}^{m-1}\hat{\phi}_A(I_{\ell,i})$, therefore $\mathbb{E}[\ell\hat{f}_{P_\ell}(A)] = \mathbb{E}[\hat{\phi}_A(I_{\ell,i})]$. Now note that if for the range space $Q = (X_{\mathcal{D},k,\ell}, R_{\mathcal{D},k,\ell})$ we consider $r_A = P_{\mathcal{D},\ell}(A)$, we have that $\frac{|X_{\mathcal{D},k,\ell} \cap r_A|}{|X_{\mathcal{D},k,\ell}|} = \mathbb{E}[\hat{\phi}_A(I_{\ell,i})]$, since $I_{\ell,i}$ is a bag of ℓ positions taken uniformly at random among all possible such bags and therefore $\mathbb{E}[\hat{\phi}_A(I_{\ell,i})]$ is the fraction of bags of ℓ positions that contain at least a position where A occurs (i.e., $\mathbb{E}[\hat{\phi}_A(I_{\ell,i})]$ is w.r.t. the uniform distribution over bags of ℓ positions). Therefore, combining Proposition 5.3.4 and Proposition 5.3.2, for the given choice of m we have that with probability $1-\delta$ it holds that $|\mathbb{E}[\ell\hat{f}_{P_\ell}(A)] - \ell\hat{f}_{P_\ell}(A)| \leq \ell\varepsilon/2, \forall A$, or, equivalently, $|\mathbb{E}[\hat{f}_{P_\ell}(A)] - \hat{f}_{P_\ell}(A)| \leq \varepsilon/2, \forall A$: we assume that this holds in the rest of the proof.

Consider a k-mer A with frequency $f_{\mathcal{D}}(A)$ in $\mathcal{D}$. From the definition of $\hat{f}_{P_\ell}(A)$, we have $\mathbb{E}[\hat{f}_{P_\ell}(A)] \leq \mathbb{E}[f_{P_\ell}(A)] = f_{\mathcal{D}}(A)$. Let $X_i = \hat{\phi}_A(I_{\ell,i})/\ell$ be the random variable taking value $1/\ell$ if the k-mer A appears at least once in the ℓ positions of $I_{\ell,i}$, and value 0 otherwise. We have that:

$$\mathbb{E}\left[\hat{f}_{P_\ell}(A)\right] = \frac{1}{m}\sum_{I_{\ell,i}\in P_\ell}\mathbb{E}\left[X_i\right] = \frac{1}{m}\sum_{I_{\ell,i}\in P_\ell}\frac{1}{\ell}\Pr\left(X_i \geq 1\right) = \left(1-\left(1-f_{\mathcal{D}}(A)\right)^{\ell}\right)/\ell \ .$$

Now consider a k-mer A with $f_{\mathcal{D}}(A) \geq 1-(1-\ell\theta)^{1/\ell}$. By the derivation above we have that $\mathbb{E}\left[\hat{f}_{P_\ell}(A)\right] \geq \theta$, and therefore its frequency $\hat{f}_{P_\ell}(A)$ in the sample P_ℓ satisfies $\hat{f}_{P_\ell}(A) \geq \theta - \varepsilon/2$, that completes the proof of the first part.

For the second part, consider a k-mer A with $f_{\mathcal{D}}(A) < \theta - \varepsilon$. By the derivation above, we have that $\mathbb{E}[\hat{f}_{P_\ell}(A)] \leq \mathbb{E}[f_{P_\ell}(A)] = f_{\mathcal{D}}(A) < \theta - \varepsilon$. Since $|\mathbb{E}[\hat{f}_{P_\ell}(A)] - \hat{f}_{P_\ell}(A)| \leq \varepsilon/2, \forall A$, we have that $\hat{f}_{P_\ell}(A) < \theta - \varepsilon/2$, which proves the second part of the result.

The third result follows from $|\mathbb{E}[\hat{f}_{P_\ell}(A)] - \hat{f}_{P_\ell}(A)| \leq \varepsilon/2$ and $\mathbb{E}[\hat{f}_{P_\ell}(A)] \leq f_{\mathcal{D}}(A)$.

The last two results follow from $|\mathbb{E}[\hat{f}_{P_\ell}(A)] - \hat{f}_{P_\ell}(A)| \leq \varepsilon/2$ and $\mathbb{E}[\hat{f}_{P_\ell}(A)] = (1-(1-f_{\mathcal{D}}(A))^{\ell})/\ell$. □

Note that from Proposition 5.3.5 the set $\{(A, f_{P_\ell}(A)) : \hat{f}_{P_\ell}(A) \geq \theta - \varepsilon/2\}$ is *almost* a ε-approximation of $FK(\mathcal{D}, k, \theta)$: in particular, there may be k-mers A for which $\mathbb{E}[\hat{f}_{P_\ell}(A)] = (1-(1-f_{\mathcal{D}}(A))^{\ell})/\ell < \theta$ while $f_{\mathcal{D}}(A) = \mathbb{E}[f_{P_\ell}(A)] \geq \theta$ and such that for the given sample P_ℓ we have $\hat{f}_{P_\ell}(A) \approx \mathbb{E}[\hat{f}_{P_\ell}(A)] - \varepsilon/2$. While this can happen, we can limit the probability of this happening by appropriately choosing ℓ, and still enjoy the reduction in sample size of the order of $\frac{\log_2 \ell}{\ell^2}$ w.r.t. Proposition 5.2.4 obtained by considering bags of bags of ℓ positions. In particular, this result allows the user to set θ, ε, δ, and ℓ to effectively find, with probability at least $1-\delta$, *all* frequent k-mers A for which $f_{\mathcal{D}}(A) \geq \theta'$ and do not report any k-mer with frequency below $\theta-\varepsilon$, while still being able to report in output almost all k-mers with frequency $\in [\theta, \theta')$. Our experimental analysis (Section 5.4) shows that in practice choosing ℓ close from below to $1/\theta$ is very effective to obtain such result. Then, the third, fourth, and fifth guarantees from Proposition 5.3.5 state that we can use the biased estimates $\hat{f}_{P_\ell}(A)$ to derive *guaranteed upper and lower bounds* to $f_{\mathcal{D}}(A)$ that will be much tighter than the one obtained using the bounds of Section 5.2.2. We will show how to obtain further improved upper and lower bounds to $f_{\mathcal{D}}(A)$ in Section 5.3.3. Such lower bounds ℓb_A can be used, for example, to prove that the set $\{(A, f_{P_\ell}(A)) : \ell b_A \geq \theta-\varepsilon\}$ enjoys the same last four guarantees from Proposition 5.3.5 while the first one holds for a $\theta' < 1-(1-\ell\theta)^{1/\ell}$; therefore, when false negatives are problematic, the set $\{(A, f_{P_\ell}(A)) : \ell b_A \geq \theta - \varepsilon\}$ can be used to obtain a different approximation of $FK(\mathcal{D}, k, \theta)$ with fewer false negatives.

Algorithm 4: SAKEIMA

Input: dataset $\mathcal{D}$, total number of k-mers $t_{\mathcal{D},k}$ in $\mathcal{D}$,
frequency threshold θ, accuracy parameter $\varepsilon \in (0, \theta)$,
confidence parameter $\delta \in (0, 1)$, integer $\ell \geq 1$.
Output: approximation $\{(A, f_A)\}$ of $FK(\mathcal{D}, k, \theta)$ with probability $\geq 1 - \delta$.

1 $m \leftarrow \left\lceil \frac{2}{(\ell\varepsilon)^2} \left(\lfloor \log_2 \min(2\ell, \sigma^k) \rfloor + \ln\left(\frac{2}{\delta}\right) \right) \right\rceil$; $\lambda \leftarrow \frac{m\ell}{t_{\mathcal{D},k}}$;
2 $T \leftarrow$ empty hash table;
3 **forall** *reads* $r_i \in \mathcal{D}$ **do**
4 **forall** $j \in [0, n_i - k]$ **do**
5 $A \leftarrow k$-mer in position j of read r_i;
6 $a \leftarrow Poisson(\lambda)$;
7 **if** $a > 0$ **then** $T[A] \leftarrow T[A] + a$;
8 $\mathcal{O} \leftarrow \emptyset$; $t \leftarrow \sum_{A \in T} T[A]$;
9 $P_\ell \leftarrow$ random partition of the occurrences in T into m bags;
10 **forall** *k-mers* $A \in T$ **do**
11 $f_A \leftarrow T[A]/t$;
12 $P_A \leftarrow$ bags of P_ℓ where A appears at least once;
13 $\hat{f}_A \leftarrow |P_A|/(m\ell)$;
14 **if** $\hat{f}_A \geq \theta - \varepsilon/2$ **then** $\mathcal{O} \leftarrow \mathcal{O} \cup (A, f_A)$;
15 **return** $\mathcal{O}$;

5.3.2 SAKEIMA: An Efficient Algorithm to Approximate Frequent k-mers

We now present our Sampling Algorithm for K-mErs approxIMAtion (SAKEIMA), that builds on Proposition 5.3.5 and efficiently samples a bag P_ℓ of bags of ℓ-positions from $\mathcal{D}$ to obtain an approximation of the set $FK(\mathcal{D}, k, \theta)$ with probability $1 - \delta$, where δ is a parameter provided by the user.

SAKEIMA is described in Algorithm 4. While SAKEIMA performs a linear scan of the input dataset, it practically reduces the number of k-mers that need to be processed with the following stategy. SAKEIMA performs a pass on the stream of k-mers appearing in $\mathcal{D}$, and for each position in the stream it draws the number a of times that the position appears in the sample P_ℓ independently at random from the Poisson distribution $Poisson(\lambda)$ of parameter $\lambda = m\ell/t_{\mathcal{D},k}$. SAKEIMA stores such values, if strictly positive, in a counting structure T (lines 3-7) that keeps, for each k-mer A, the total number of occurrences of A in the sample P_ℓ. We implicitly assume that obtaining A, in line 5, is not costly, while it is expensive to *count it*, i.e. inserting it in T, as done in line 7 when $a > 0$. Depending on the value of λ, an alternative strategy may be to sample the number of positions to *skip*, reducing the number of draws from the Poisson distribution. Then, we note that $t_{\mathcal{D},k}$ can be

computed with a very quick linear scan of the dataset, where n_i is computed for every $r_i \in \mathcal{D}$ without extracting and processing (e.g., inserting or updating information for) k-mers; in alternative a lower bound to $t_{\mathcal{D},k}$ can be used, simply resulting in a number of samples higher than needed. For each k-mer A appearing at least once in the sample, the unbiased estimate f_A is computed in line 11 as the number $T[A]$ of occurrences of A in the sample P_ℓ divided by the total number of positions in the sample t. The biased estimate $\hat{f}_A$ can be computed partitioning the $T[A]$ occurrences of A into m bags $I_{\ell,0}, \ldots, I_{\ell,m-1}$; $\hat{f}_A$ is then simply the ratio between the number of bags where A appears at least once and $m\ell$. We describe a more efficient way of computing such biased estimate at the end of this section. Then `SAKEIMA` flags A as frequent if $\hat{f}_A \geq \theta - \varepsilon/2$ (line 14) and, in this case, the couple (A, f_A) is added to the output set $\mathcal{O}$ (line 15), since f_A is the best (and unbiased) estimate to $f_{\mathcal{D}}(A)$.

Note that `SAKEIMA` does not sample m bags of *exactly* ℓ positions each, since the number of occurrences of each position in $\mathcal{D}$ in the sample P_ℓ is sampled independently from a Poisson distribution, even if the expected number of total occurrences sampled from the algorithm is $m\ell$. However, the independent Poisson distributions used by `SAKEIMA` provide an accurate approximation of the random sampling of *exactly* $m\ell$ positions used in the analysis of Section 5.3.1. In particular, this holds when one focuses on the events of interests for our approximation of Section 5.3.1 (e.g., the event "there exists a k-mer A such that $|\mathbb{E}[\hat{f}_{P_\ell}(A)] - \hat{f}_{P_\ell}(A)| > \varepsilon/2$"). In fact, a simple adaptation of a known result (Corollary 5.11 of (Mitzenmacher and Upfal, 2017)) on the relation between sampling with replacement and the use of independent Poisson distributions gives the following.

Proposition 5.3.6. *Let E be an event whose probability is either monotonically increasing or monotonically decreasing in the number of sampled positions. If E has probability p when the independent Poisson distributions are used, then E has probability at most $2p$ when the sampling with replacement is used.*

As a simple corollary, the output $\mathcal{O}$ features the guarantees of Proposition 5.3.5 with probability $\geq 1 - \delta'$, with $\delta' = 2\delta$.

The technique we just described can be used to avoid the exact computation of $\hat{f}_A$, which requires to maintain and update the counters for the m buckets; in fact, we can approximate the number of occurrences of a k-mer A, appearing $T[A]$ times in the random sample of `SAKEIMA` into a given bucket as a sample from $Poisson(T[A]/m)$. This means that the number of buckets where A will be inserted *at least once* is well approximated by a sample from $Binomial(m, 1-e^{-T[A]/m})$, which models the number of successes in m independent trials with probability of success $1-e^{-T[A]/m}$. Due to this second Poisson approximation, we obtain that the output $\mathcal{O}$ provides the guarantees of Proposition 5.3.5 with probability $\geq 1-\delta''$, with $\delta'' = 4\delta$. In terms of Algorithm 4, such modification simply requires to substitute $\frac{2}{\delta}$ with $\frac{4}{\delta}$ in line 1, to remove line 9, and to substitute lines 12-13 with "$\hat{f}_A \leftarrow Binomial(m, 1-e^{-T[A]/m})/(m\ell)$". This also allows to efficiently compute multiple values of $\hat{f}_A$, corresponding to different values

of ℓ, by simply taking samples from binomial distributions of different appropriate parameters. (In particular, if one samples a total t of k-mers, then the value m to be used for both parameters of the binomial distribution is t/ℓ.) The next section shows why this is useful.

5.3.3 Improved Lower and Upper Bounds to k-mer Frequencies

Note that Proposition 5.3.5 guarantees that we can obtain upper and lower bounds to $f_{\mathcal{D}}(A)$ for every $A \in FK(\mathcal{D}, k, \theta)$ from the sample of bags of ℓ positions. These bounds are meaningful only in specific ranges of the frequencies; for example, the lower bound from the third guarantee in Proposition 5.3.5 is meaningful when the frequency of A is fairly low, i.e $f_{\mathcal{D}}(A) \approx 1/\ell$, while for very frequent k-mers they could be a multiplicative factor $1/\ell$ away from than the correct value. For example, if a k-mer is very frequent and appears in all bags of ℓ k-mers in a sample S, its corresponding lower bound is still only $1/\ell - \varepsilon/2$.

However, Proposition 5.3.5 can be generalized to obtain tighter upper and lower bounds to the frequency of all k-mers. For given ℓ, ε, and δ, let m as given in Proposition 5.3.5. Note that the total number of k-mer's positions in the sample P_ℓ is $m\ell$. Let $\mathcal{L}$ be a set of integer values $\mathcal{L} = \{\ell_i\}$ with $\ell_i \in [1, m\ell], \forall i = 0, \ldots, |\mathcal{L}| - 1$. Now, for every $\ell_i \in \mathcal{L}$, we can partition the *same* $m\ell$ k-mers that are in P_ℓ into $m_i = m\ell/\ell_i$ partitions having size ℓ_i. Let P_{ℓ_i} be such a random partition of such positions into m_i bags of ℓ_i positions each. Note that each P_{ℓ_i} is a "valid" sample (i.e., a sample of independent bags of positions, each obtained by uniform sampling with replacement) for Proposition 5.3.5, even if the P_{ℓ_i}'s are not independent. From each P_{ℓ_i}, we define a maximum deviation ε_i from Proposition 5.3.5 as

$$\varepsilon_i = \frac{1}{\ell_i}\sqrt{\frac{2}{m_i}\left(\lfloor \log_2(\min(2\ell_i, \sigma^k))\rfloor + \ln\left(\frac{|\mathcal{L}|}{\delta}\right)\right)}.$$

We have the following result.

Proposition 5.3.7. *With probability at least $1-\delta$, for all k-mers A simultaneously and for all the random partitions induced by $\mathcal{L}$ it holds*

- $f_{\mathcal{D}}(A) \geq \max\{\hat{f}_{P_{\ell_i}}(A) - \varepsilon_i/2 : i \in [0, |\mathcal{L}| - 1]\}$;
- $f_{\mathcal{D}}(A) \geq \max\{1-(1-\ell_i(\hat{f}_{P_{\ell_i}}(A)-\varepsilon_i/2))^{1/\ell_i} : i \in [0, |\mathcal{L}|-1], \hat{f}_{P_{\ell_i}}(A)-\varepsilon_i/2 \geq 0\}$;
- $f_{\mathcal{D}}(A) \leq \min\{1 - (1 - \ell_i(\hat{f}_{P_{\ell_i}}(A) + \varepsilon_i/2))^{1/\ell_i} : i \in [0, |\mathcal{L}| - 1], \hat{f}_{P_{\ell_i}}(A) + \varepsilon_i/2 \leq 1/\ell_i\}$.

Proof. Combining Proposition 5.3.4 and Proposition 5.3.2 and by union bound on the $|\mathcal{L}|$ values of i, we have that with probability $1-\delta$ it holds that $|\mathbb{E}[\hat{f}_{P_{\ell_i}}(A)]-\hat{f}_{P_{\ell_i}}(A)| \leq$

$\varepsilon_i/2, \forall A$ and $\forall i \in [0, |\mathcal{L}| - 1]$: we assume that this holds in the rest of the proof. To prove the lower bound, note that since $\mathbb{E}[\hat{f}_{P_{\ell_i}}(A)] = (1 - (1 - f_{\mathcal{D}}(A))^{\ell_i})/\ell_i$, from the above we have that

$$(1 - (1 - f_{\mathcal{D}}(A))^{\ell_i})/\ell_i \geq \hat{f}_{P_{\ell_i}}(A) - \varepsilon_i/2$$

that is equivalent to

$$f_{\mathcal{D}}(A) \geq 1 - (1 - \ell_i(\hat{f}_{P_{\ell_i}}(A) - \varepsilon_i/2))^{1/\ell_i}$$

when $\hat{f}_{P_{\ell_i}}(A) - \varepsilon_i/2 \geq 0$. The proof of the upper bound is analogous. □

In our experiments, we use $\mathcal{L} = \{\ell_i\}$ with $\ell_i = \ell/2^i, \forall i \in [0, \lfloor \log_2 \ell \rfloor - 1]$; in this case, note that $P_{\ell_0} = P_\ell$. Using this scheme, we can compute upper and lower bounds for k-mers having frequencies of many different orders of magnitude, but any (application dependent) distribution can be specified by the user. Then, these upper and lower bounds can be used to obtain different approximations of $FK(\mathcal{D}, k, \theta)$ with different guarantees. For example, by reporting all k-mers (and their frequencies) that have an upper bound $\geq \theta$, we have an approximation that guarantees that all k-mers A with $f_{\mathcal{D}}(A) \geq \theta$ are in the approximation.

5.4 Experimental Results

In this section we present the results of our experimental evaluation for `SAKEIMA`. Section 5.4.1 describes the datasets, our implementation for `SAKEIMA`[1], and the baseline for comparisons. In Section 5.4.2, we report the results for computing the approximation of the frequent k-mers using `SAKEIMA`. Section 5.4.3 reports the results of using our approximation to compute abundance-based and presence-based distances between metagenomic datasets.

5.4.1 Datasets and Implementation

We considered six datasets from the Human Microbiome Project (HMP)[2], one of the largest publicly available collection of metagenomic datasets from high-throughput sequencing. In particular, we selected the three largest datasets of `stool` and the three largest of `tongue dorsum` (Table 5.1). These datasets constitute the most challenging instances, due to their size, and provide a test case with different degrees of similarities among datasets. We implemented `SAKEIMA` in `C++` as a modification of Jellyfish (Marçais and Kingsford, 2011) (the version we used is 2.2.10[3]), a very pop-

[1] Available at https://github.com/VandinLab/SAKEIMA
[2] https://hmpdacc.org/HMASM
[3] https://github.com/gmarcais/Jellyfish

ular and efficient algorithm for exact k-mer counting. Doing so, our algorithm enjoys the succinct counting data structure provided by Jellyfish publicly available implementation. We remark that our sampling-based approach can be used in combination with any other highly tuned method available for exact, approximate, and parallel k-mer counting. For this reason, we only compare `SAKEIMA` with the exact counting performed by Jellyfish, since they share the underlying characteristics, allowing us to evaluate the impact of `SAKEIMA`'s sampling strategy.

For running time and memory we computed the average from 10 runs. When comparing Jellyfish and `SAKEIMA` using 1 worker, we show the CPU time, while when using multiple threads we show the overall running time. We did not include the time to compute $t_{\mathcal{D},k}$ in our experiments since we assume it is provided in input (for example, computed while the dataset of read is created). In cases when it is not known in advance, $t_{\mathcal{D},k}$ can be computed by simply scanning all the k-mers without counting them. We computed the time required for this task for the datasets we consider and it was always small (i.e., always less than 175 seconds with 1 worker, and than 70 seconds with 32 workers) compared to the time for counting k-mers.

For the computation of the abundance-based distances from the k-mer counts of two dataset, we implemented in `C++` a simple algorithm that loads the counts of one dataset in main memory and then performs one pass on the counts of the other dataset, producing the distances in output. We executed all our experiments on the same machine with 512 GB of RAM and 2.30 GHz Intel Xeon CPUs (with 64 cores in total), compiling both implementations with `GCC 8`. `SAKEIMA` can be used in combination with more efficient algorithms and implementations for the computation of these (and other) distances (Benoit et al., 2016), resulting in speed-ups analogous to the ones we present below. For all the experiments of `SAKEIMA`, given θ and a dataset $\mathcal{D}$, we fixed the parameters $\delta = 0.1$, $\varepsilon = \theta - 2/t_{\mathcal{D},k}$, and we fix $\ell = \lfloor 0.9/\theta \rfloor$.

5.4.2 Approximation of the Frequent k-mers

We fixed $k = 31$, and we compared `SAKEIMA` with the exact counting of all k-mers (from Jellyfish) in terms of: (i) running time, including, for both algorithms, the time required to write the output on disk; (ii) memory requirement. We also assessed the accuracy of the output of `SAKEIMA`.

Figure 5-2 shows the average running times and peak memory as function of θ, using 1 worker. Note that for the exact counting algorithm these metrics do not depend on θ, since it always counts all k-mers. `SAKEIMA` is always faster than the exact counting, with a difference that increases when θ increases and a speed-up around 2 even for $\theta = 2 \cdot 10^{-8}$. The memory requirement of `SAKEIMA` reduces when θ increases, and for $\theta = 2 \cdot 10^{-8}$ it is half of the memory required by the exact counting. This is due to `SAKEIMA`'s sample size being much smaller than the dataset size (Figure 5-2(d)), therefore a large portion of extremely low frequency k-mers are naturally left out from the random sample and do not need to be accounted for in the

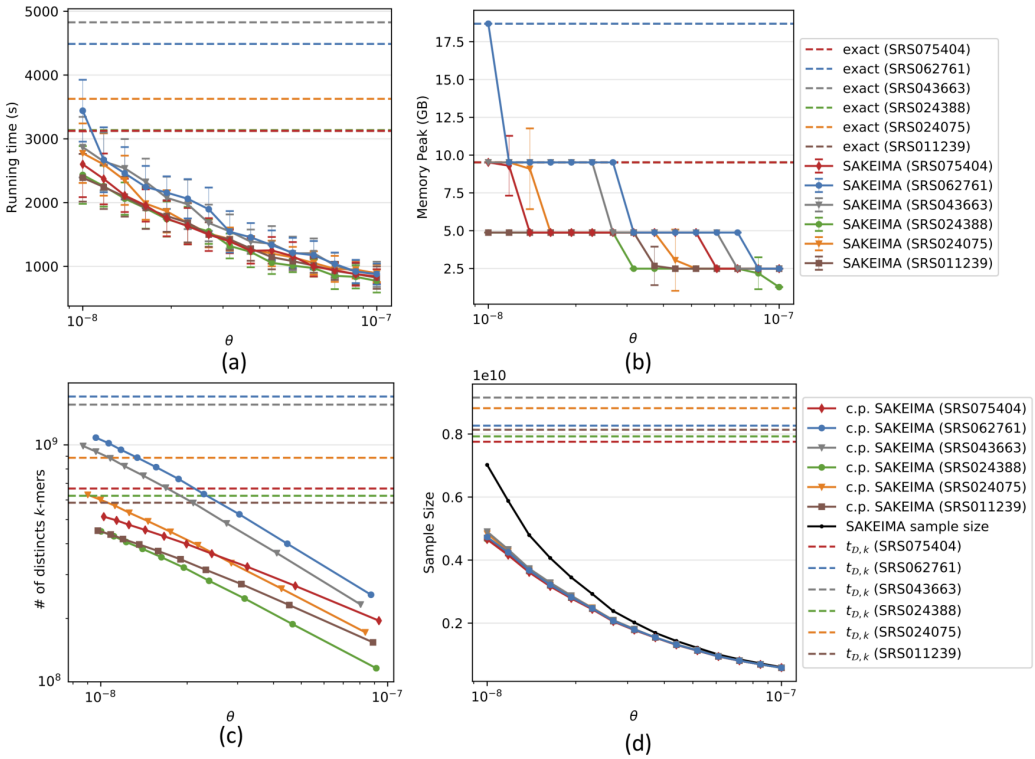


Figure 5-2: Running time, memory requirements, and number of distinct k-mers counted, for SAKEIMA and exact counting as function of θ. (a) Running time (average ± 2 standard deviations from 10 runs). (b) Memory requirement (the standard deviation is not shown when all the 10 runs have the same peak memory). (c) Number of distinct k-mers counted. (d) Sample sizes of SAKEIMA, total size $t_{\mathcal{D},k}$ of the datasets, and number (c.p.) of dataset's distinct covered positions (i.e., included in SAKEIMA's sample), as function of θ.

counting data structure, as confirmed by counting the number of *distinct* k-mers that are inserted in the counting data structure by the two algorithms (Figure 5-2(c)). (The difference between the memory requirement and the number of distinct k-mers is given by Jellyfish's strategy to doubles the size of the counting data structure when it is full.)

Figure 5-3 shows the average running times of SAKEIMA and Jellyfish as function of θ and the number of workers w for counting k-mer from dataset SRS043663. The memory used by both approaches does not depend on w, therefore it is the same of Figure 5-2. We can see that increasing w reduces the running time of both approaches, and that the relative improvements provided by the sampling strategy of SAKEIMA is mantained. This shows that SAKEIMA is well suited to be combined with parallel approaches.

In terms of quality of the approximation, the output of SAKEIMA satisfied the

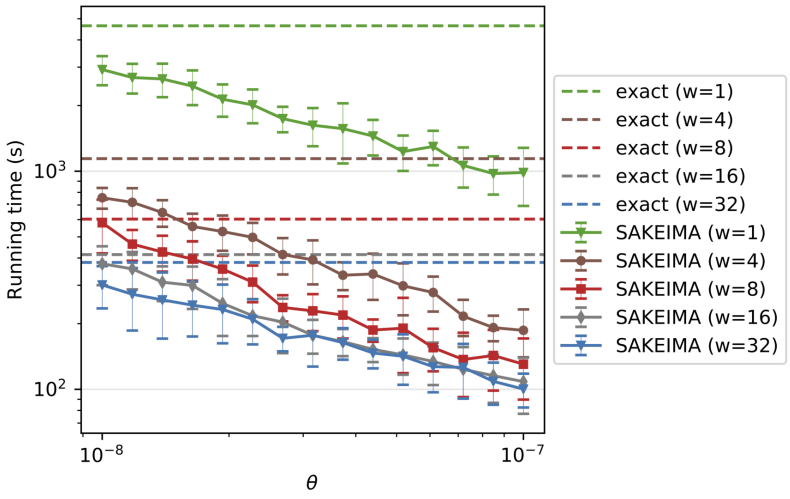


Figure 5-3: Running time for `SAKEIMA` and exact counting for dataset `SRS043663`, as function of θ and the number of workers w.

guarantees given by Proposition 5.3.5 for all runs of our experiments, therefore with probability higher than $1 - \delta$. While `SAKEIMA` may incur in false negatives, its false negative ratio (i.e., the fraction of k-mers in $FK(\mathcal{D}, k, \theta)$ not reported by `SAKEIMA`) is always $\leq 3 \cdot 10^{-4}$ (Figure 5-4(a)), even if the sampling technique of Section 5.3.1 does not provide rigorous guarantees on such quantity. Therefore `SAKEIMA` is very effective in reporting almost all frequent k-mers. As mentioned in Section 5.3.3, `SAKEIMA` can be easily modified so to report all frequent k-mers in output, even if at the cost of reporting also more k-mers with frequency between $\theta - \varepsilon$ and θ. In addition, the estimated frequencies f_A reported by `SAKEIMA` are always close to the true values $f_{\mathcal{D}}(A)$, with a small maximum deviation $|f_A - f_{\mathcal{D}}(A)|$ (Figure 5-4(b)), and an even smaller average deviation (Figure 5-4(c)). In addition, the upper and lower bounds computed as in Section 5.3.3 provide small confidence intervals always containing the value $f_{\mathcal{D}}(A)$ (e.g., Figure 5-4(d) for dataset `SRS062761`), and could be used to obtain sets of k-mers with various guarantees from the sample used by `SAKEIMA`.

5.4.3 Application to Metagenomics: Computation of Ecological Distances

We evaluate the use of `SAKEIMA` to speed up the computation of commonly used k-mer based ecological distances (Benoit et al., 2016) between datasets of Next-Generation Sequencing (NGS) reads. We present results for the Bray-Curtis distance; analogous results hold for other distances (see Section 5.5).

We first investigated how the distances change when those are computed considering only the *frequent* k-mers (w.r.t. a frequency threshold θ) instead that the full

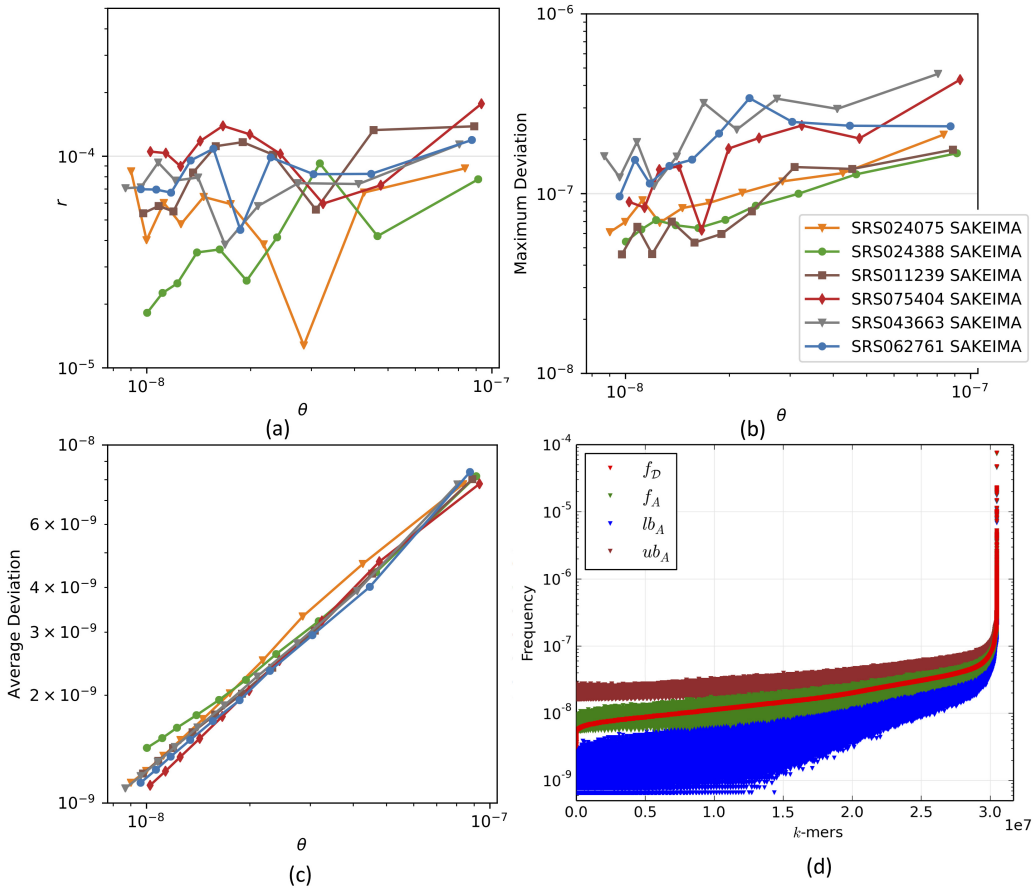


Figure 5-4: Quality of the approximation of $FK(\mathcal{D}, k, \theta)$ produced by `SAKEIMA`. (a) False negative rate, i.e., the fraction r of k-mers in $FK(\mathcal{D}, k, \theta)$ not reported by `SAKEIMA`. (b) Maximum deviation $|f_A - f_{\mathcal{D}}(A)|$ of the estimates reported by `SAKEIMA` for various θ. (c) Average value of $|f_A - f_{\mathcal{D}}(A)|$ for the k-mers A reported by `SAKEIMA` for various θ. (d) Frequencies and bounds for dataset `SRS062761` and $\theta = 10^{-8}$ shown for k-mers sorted in increasing order of exact frequencies. Red: exact frequencies $f_{\mathcal{D}}(A)$. Green: estimate f_A of $f_{\mathcal{D}}(A)$ from `SAKEIMA`. Blue: lower bound lb_A to $f_{\mathcal{D}}(A)$ from `SAKEIMA`. Brown: upper bound ub_A to $f_{\mathcal{D}}(A)$ from `SAKEIMA`.

spectrum of k-mers appearing in the data. Therefore, given a pair of datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ and θ, we computed the sets $\mathcal{O}_1 = FK(\mathcal{D}_1, k, \theta)$ and $\mathcal{O}_2 = FK(\mathcal{D}_2, k, \theta)$ using Jellyfish and then computed a generalized version of the distances for all pairs of datasets we used for our experiments. For the Bray-Curtis distance, this generalization is defined as:

$$BC(\mathcal{D}_1, \mathcal{D}_2, \mathcal{O}_1, \mathcal{O}_2) = 1 - 2\frac{\sum_{A \in \mathcal{O}_1 \cap \mathcal{O}_2} \min\{o_{\mathcal{D}_1}(A), o_{\mathcal{D}_2}(A)\}}{\sum_{A \in \mathcal{O}_1} o_{\mathcal{D}_1}(A) + \sum_{A \in \mathcal{O}_2} o_{\mathcal{D}_2}(A)} \ .$$

Note that when $\theta \leq 10^{-10}$ then $FK(\mathcal{D}, k, \theta)$ coincides with the set of *all* k-mers, for any of the datasets we tested. The results (Figure 5-5(a)) show that for θ up to 5×10^{-8} the values of the distances are fairly stable and therefore one can use only frequent k-mers for such values of θ to compute the distances, and that for θ up to 10^{-7} the relation between distances of different pairs of datasets are almost always conserved. We underline that the exact counting approach needs to count *all* the k-mers and only afterwards can filter the infrequent ones before writing them to disk to compute $FK(\mathcal{D}, k, \theta)$. We then used `SAKEIMA` to extract approximations (of k-mers and their frequencies) of $FK(\mathcal{D}_1, k, \theta)$ and $FK(\mathcal{D}_2, k, \theta)$ and used such approximations to compute the distances among datasets (Figure 5-5(b)). Strikingly, the distances computed from the output of `SAKEIMA` are very close to their exact variant (Figure 5-5(c)). Interestingly this holds also for the Jaccard distance, a presence-based distance that does not depend neither on k-mer abundances nor on k-mer ranking by frequencies.

We then compared, for different values of θ, the total running time required to compute the approximations of the frequent k-mers using `SAKEIMA` for all datasets in Table 5.1 and all distances among such datasets using `SAKEIMA` approximations with the running time required when the exact counting algorithm is used for the same tasks. `SAKEIMA` reduces the computing time by more than 75% (Figure 5-5(d)). This result comes from both the efficiency of `SAKEIMA` and from the fact that by focusing on the the most frequent k-mers we greatly reduce the number of distinct k-mers that need to be processed for computing the distances. Therefore `SAKEIMA` can be used for a very fast comparison of metagenomic datasets while preserving the ability of distinguishing similar datasets from different ones.

5.5 Proofs and Additional Results

In this Section we provide more details and missing proofs for some of the results presented in this Chapter.

A first VC Dimension-based Bound

In this section we prove Proposition 5.2.4, which gives an improved bound on the sample size required to obtain a rigorous approximation to $FK(\mathcal{D}, k, \theta)$ w.r.t. the

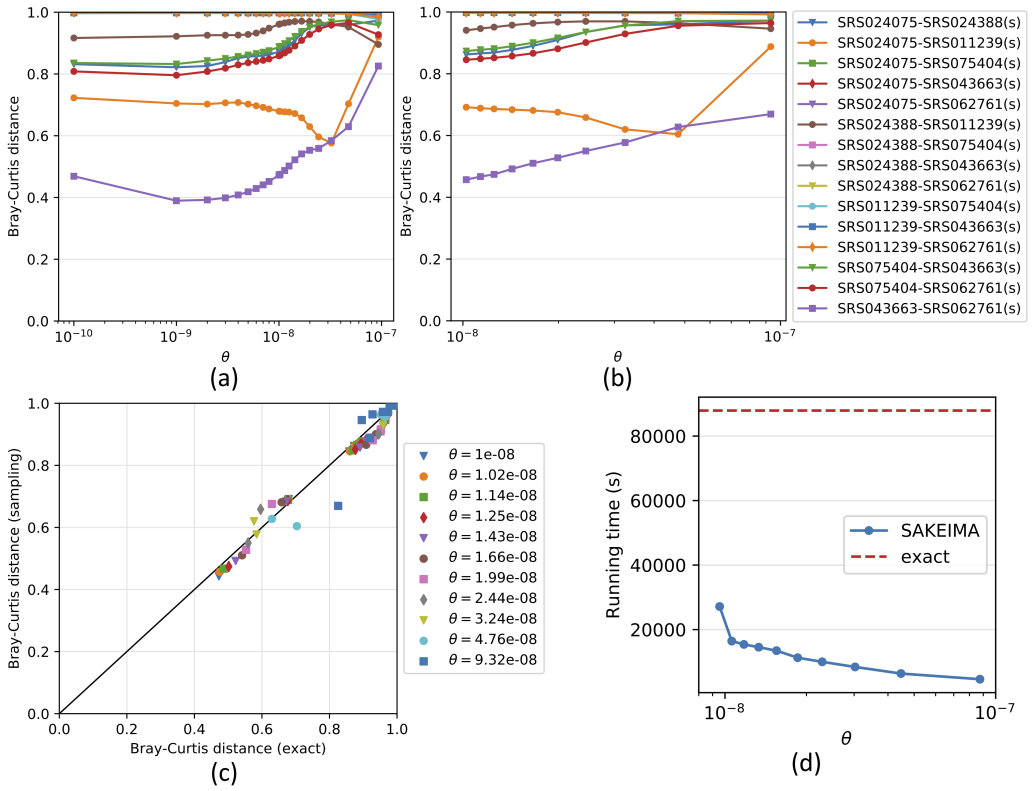


Figure 5-5: Results for Bray-Curtis (BC) distances of metagenomic datasets. (a) BC distance computed using k-mers with frequency $\geq \theta$. (b) BC distances computed using the approximation of k-mers with frequency $\geq \theta$ from `SAKEIMA`. (c) Comparison of the BC distance using all k-mers with exact counts and the approximation of frequent k-mers by `SAKEIMA`. (d) Total time required by `SAKEIMA` and the exact approach to find frequent k-mers and compute all distances between datasets as a function of θ.

Table 5.1: Datasets for our experimental evaluation. For each dataset $\mathcal{D}$ the table shows: the dataset name and site (`(s)` for `stool`, `(t)` for `tongue dorsum`); the total number $t_{\mathcal{D},k}$ of k-mers ($k = 31$) in $\mathcal{D}$; the number $|\mathcal{D}|$ of reads it contains; the maximum read length $\max_{n_i} = \max_i\{n_i | r_i \in \mathcal{D}\}$; the average read length $\text{avg}_{n_i} = \sum_{i=0}^{n-1} n_i/n$.

dataset	$t_{\mathcal{D},k}$	$\lvert\mathcal{D}\rvert$	$\max_{n_i}$	avg_{n_i}
`SRS024388(s)`	$7.92 \cdot 10^9$	$1.20 \cdot 10^8$	102	97.21
`SRS011239(s)`	$8.13 \cdot 10^9$	$1.24 \cdot 10^8$	102	96.69
`SRS024075(s)`	$8.82 \cdot 10^9$	$1.38 \cdot 10^8$	96	94.88
`SRS075404(t)`	$7.75 \cdot 10^9$	$1.22 \cdot 10^8$	102	94.51
`SRS062761(t)`	$8.26 \cdot 10^9$	$1.18 \cdot 10^8$	101	101.00
`SRS043663(t)`	$9.15 \cdot 10^9$	$1.31 \cdot 10^8$	101	101.00

one given by 5.2.3. In our analysis we use the Vapnik-Chervonenkis (VC) dimension (Vapnik, 1998), a statistical learning concept that measures the expressivity of a family of binary functions.

We now define the range space associated to k-mers and derive an upper bound to its VC dimension.

Definition 5.5.1. Let $\mathcal{D}$ be a bag of n reads and let $k > 0$ be an integer. For any k-mer A, let $P_{\mathcal{D},k}(A)$ be the set of elements of $P_{\mathcal{D},k}$ corresponding to the occurrences of A in $\mathcal{D}$. We define the range space $Q = (X_{\mathcal{D},k}, R_{\mathcal{D},k})$ associated to the k-mers in $\mathcal{D}$ as follows:

- $X_{\mathcal{D},k}$ is the set of all occurrences of k-mers in $\mathcal{D}$, that is: $X_{\mathcal{D},k} = P_{\mathcal{D},k}$;
- $R_{\mathcal{D},k} = \{P_{\mathcal{D},k}(A) | A \in \Sigma^k\}$.

Note that for any A, if we consider $r = P_{\mathcal{D},k}(A) \subseteq P_{\mathcal{D},k}$ we have $|X_{\mathcal{D},k} \cap r|/|X_{\mathcal{D},k}| = f_{\mathcal{D}}(A)$. Therefore, by taking $\mathcal{D} = X_{\mathcal{D},k}$ and $R_X = R_{\mathcal{D},k}$ in Definition 5.3.1, we have that an ε-approximation B of $X_{\mathcal{D},k}$ guarantees that $|f_{\mathcal{D}}(A) - f_B(A)| \leq \varepsilon/2$.

A trivial upper bound (Shalev-Shwartz and Ben-David, 2014) to the VC dimension v of the range space $Q = (X_{\mathcal{D},k}, R_{\mathcal{D},k})$ is given by $v \leq \lfloor \log_2 |R_{\mathcal{D},k}| \rfloor = \lfloor \log_2 \sigma^k \rfloor$. However, we are able to precisely characterize $VC(Q)$, that is instrumental in obtaining an improved bound on the number of samples required for a ε-approximation.

Proposition 5.5.2. *Let $\mathcal{D}$ be a bag of n reads, $k > 0$ an integer, and $Q = (X_{\mathcal{D},k}, R_{\mathcal{D},k})$ be the corresponding range space. Then the VC dimension $VC(Q)$ of Q is 1.*

Proof. The proof is by contradiction. Assume that $VC(Q) = v' > 1$: therefore there exists a set $X \subseteq X_{\mathcal{D},k}$ with $|X| = v'$ that can be shattered by $R_{\mathcal{D},k}$. In order to be shattered, there should exist at least $2^{v'}$ k-mers $A_1, A_2, \ldots, A_{2^{v'}}$ such that the projection of their corresponding ranges on X gives all subsets of X. Consider two subsets X', X'' of X for which $X' \neq X''$ and $X' \cap X'' \neq \emptyset$. Since X' and X'' must be in the projection of the ranges corresponding to $A_1, A_2, \ldots, A_{2^{v'}}$ on X, there must exist two distinct k-mers A_i and A_j for which $P_{\mathcal{D},k}(A_i) = X'$ and $P_{\mathcal{D},k}(A_j) = X''$. This is a contradiction, since if $X' \cap X'' \neq \emptyset$, then each position in $X' \cap X''$ must be the starting position for the two distinct k-mers A_i and A_j, while a position can be the starting position for only one k-mer. □

Proposition 5.2.4 follows directly from Proposition 5.3.2 and from Proposition 5.5.2; it provides a VC dimension-based bound on the number of samples required to obtain an ε-approximation of $FK(\mathcal{D}, k, \theta)$.

Frequency Histograms of 31-mers

We show in Figure 5-6 the exact frequency histograms we computed with Jellyfish of the k-mers (with $k = 31$) for all the datasets we considered in our experiments.

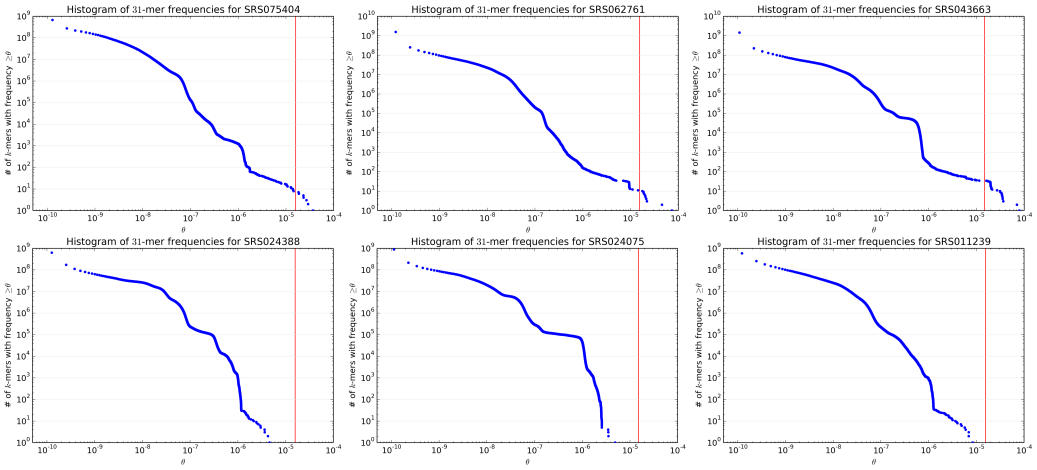


Figure 5-6: Histograms of the exact frequencies of the datasets we tested. The vertical red line is drawn in correspondence of a lower bound to $\theta - \varepsilon/2 = \frac{1}{2}\sqrt{\frac{2}{t_{\mathcal{D},k}}\left(1 + \log(\frac{1}{\delta})\right)}$ (with $\delta = 0.05$), that is the lowest achievable frequency threshold using the results of Section 5.5.

For every dataset we computed $\frac{1}{2}\sqrt{\frac{2}{t_{\mathcal{D},k}}\left(1 + \log(\frac{1}{\delta})\right)}$ (with $\delta = 0.05$), that is a lower bound to the frequency threshold $\theta - \varepsilon/2$ (drawn in the plots with red vertical lines) that can be obtained from the results of Section 5.5.

Distances between Datasets of Reads

In our experimental evaluation we considered three abundance-based distances and one presence-based distance commonly used to compare metagenomic datasets (Benoit et al., 2016), and generalized them to the scenario in which only a set of all k-mers are observed. Let $\mathcal{O}$ be a subset of the set of all possible k-mers. Define the indicator functions:

- $f_{\mathcal{D},\mathcal{O}}(A) = f_{\mathcal{D}}(A)$ if $A \in \mathcal{O}$, $f_{\mathcal{D},\mathcal{O}}(A) = 0$ otherwise; and
- $o_{\mathcal{D},\mathcal{O}}(A) = o_{\mathcal{D}}(A)$ if $A \in \mathcal{O}$, $o_{\mathcal{D},\mathcal{O}}(A) = 0$ otherwise.

Given two datasets $\mathcal{D}_1, \mathcal{D}_2$, let $\mathcal{O}_1$ and $\mathcal{O}_2$ be the k-mers observed for $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively. We considered the following distances:

- the Bray-Curtis distance:

$$BC(\mathcal{D}_1, \mathcal{D}_2, \mathcal{O}_1, \mathcal{O}_2) = 1 - 2\frac{\sum_{A \in \Sigma^k} \min\{o_{\mathcal{D}_1,\mathcal{O}_1}(A), o_{\mathcal{D}_2,\mathcal{O}_2}(A)\}}{\sum_{A \in \Sigma^k} o_{\mathcal{D}_1,\mathcal{O}_1}(A) + \sum_{A \in \Sigma^k} o_{\mathcal{D}_2,\mathcal{O}_2}(A)};$$

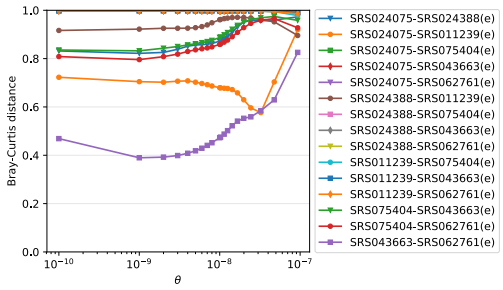


Figure 5-7: Bray-Curtis distance on the exact set of frequent k-mers.

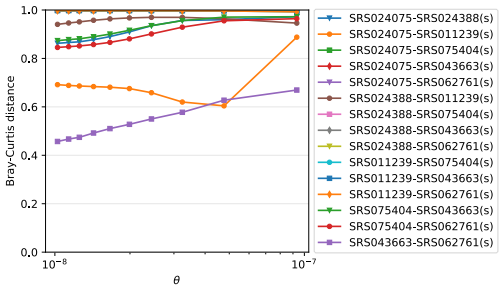


Figure 5-8: Bray-Curtis distance on output of `SAKEIMA`.

- the Whittaker distance:

$$W_t(\mathcal{D}_1, \mathcal{D}_2, \mathcal{O}_1, \mathcal{O}_2) = \frac{1}{2} \sum_{A \in \Sigma^k} |f_{\mathcal{D}_1,\mathcal{O}_1}(A) - f_{\mathcal{D}_2,\mathcal{O}_2}(A)|;$$

- the Chord distance:

$$C_h(\mathcal{D}_1, \mathcal{D}_2, \mathcal{O}_1, \mathcal{O}_2) = \sqrt{2 - 2 \sum_{A \in \Sigma^k} \frac{o_{\mathcal{D}_1,\mathcal{O}_1}(A) o_{\mathcal{D}_2,\mathcal{O}_2}(A)}{\sqrt{\sum_{A \in \Sigma^k} o_{\mathcal{D}_1,\mathcal{O}_1}(A)^2} \sqrt{\sum_{A \in \Sigma^k} o_{\mathcal{D}_2,\mathcal{O}_2}(A)^2}}};$$

- the Jaccard distance:

$$J_c(\mathcal{D}_1, \mathcal{D}_2, \mathcal{O}_1, \mathcal{O}_2) = 1 - \frac{|\mathcal{O}_1 \cap \mathcal{O}_2|}{|\mathcal{O}_1 \cup \mathcal{O}_2|}.$$

For the Jaccard distance, we considered only k-mers appearing at least twice in the datasets, since k-mers with count 1 often represents sequencing errors and greatly affect the accuracy of presence-based distances, such as the Jaccard distance.

In Figures 5-7 - 5-14 we show the distances computed using the exact sets of frequent k-mers, for different values of θ, and using the corresponding approximated sets given by `SAKEIMA`. Figures 5-15 - 5-18 directly compare them.

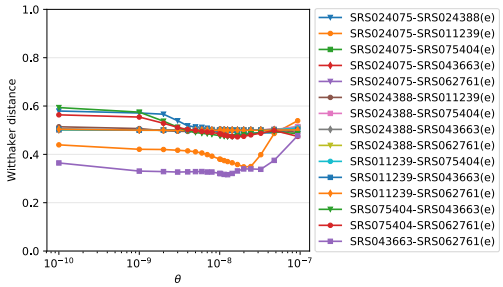


Figure 5-9: Whittaker distance on the exact set of frequent k-mers.

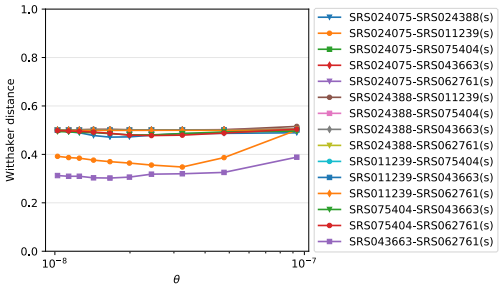


Figure 5-10: Whittaker distance on output of `SAKEIMA`.

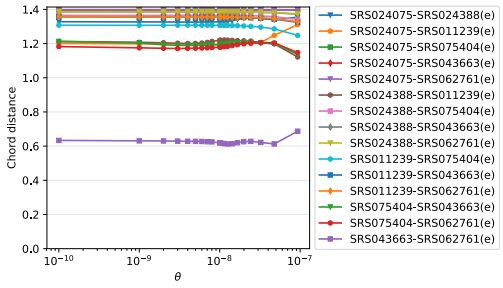


Figure 5-11: Chord distance on the exact set of frequent k-mers.

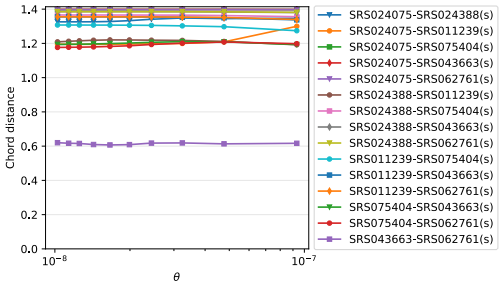


Figure 5-12: Chord distance on output of `SAKEIMA`.

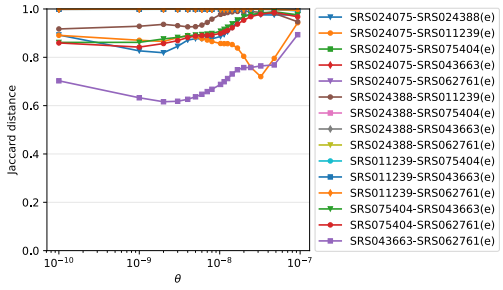


Figure 5-13: Jaccard distance on the exact set of frequent k-mers.

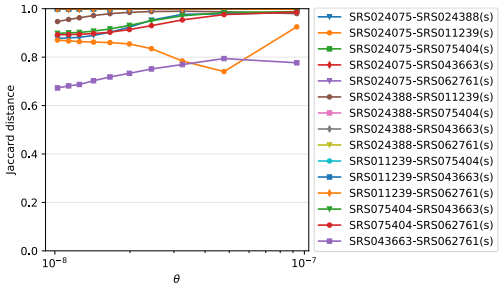


Figure 5-14: Jaccard distance on output of `SAKEIMA`.

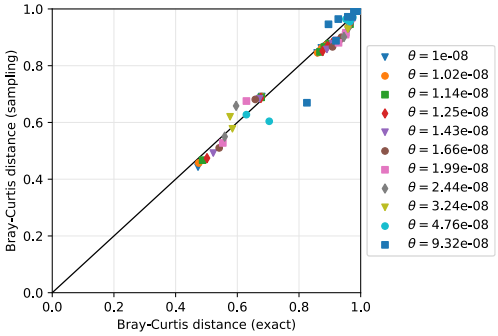


Figure 5-15: Bray-Curtis distance.

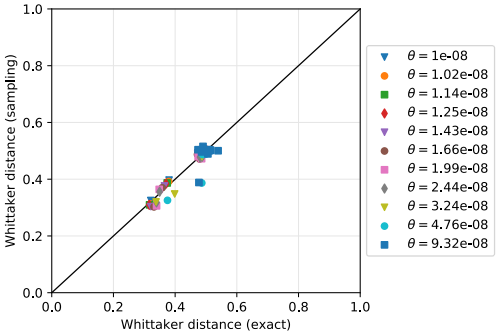


Figure 5-16: Whittaker distance.

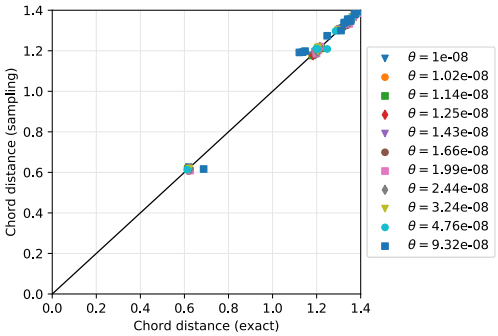


Figure 5-17: Chord distance.

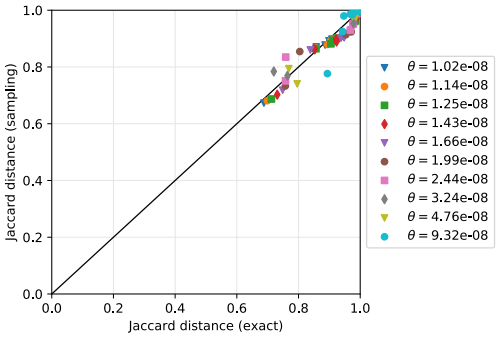


Figure 5-18: Jaccard distance.

Chapter 6

Monte Carlo Rademacher Averages for Poset Families and Approximate Pattern Mining

6.1 Introduction

Pattern Mining is a key sub-area of Knowledge Discovery from data, with a large number of variants tailored to applications ranging from market basket analysis, to spam detection, and to recommendation systems.

In this Chapter we are interested in the analysis of *samples* for Pattern Mining. There are two meanings of "sample" in this context, but, as we now argue, they are really two sides of the same coin, and our methods work for both sides.

The first meaning is *sample* as a *small random sample of a large dataset*: since mining patterns becomes more expensive as the dataset grows, it is reasonable to mine only a small random sample that fits into the main memory of the machine. Recently, this meaning of sample as "sample-of-the-dataset" has been used also to enable interactive data exploration using progressive algorithms for pattern mining (Servan-Schreiber et al., 2018a). The patterns obtained from the sample are an *approximation* of the exact collection, due to the noise introduced by the sampling process. To obtain desirable probabilistic guarantees on the quality of the approximation, one must study the *trade-off between the size of the sample and the quality of the approximation*. Many works have progressively obtained better characterizations of the trade-off using advanced probabilistic concepts (Toivonen, 1996; Chakaravarthy et al., 2009; Riondato and Upfal, 2014, 2015; Riondato and Vandin, 2018; Servan-Schreiber et al., 2018a). Recent methods (Riondato and Upfal, 2014, 2015; Riondato and Vandin, 2018; Servan-Schreiber et al., 2018a) use VC-dimension, pseudodimension, and Rademacher averages (Bartlett and Mendelson, 2002; Koltchinskii and Panchenko, 2000), key concepts from statistical learning theory (Vapnik, 1998) (see also Sections 6.2 and 6.3.2), because they allow to obtain uniform (i.e., simultaneous) probabilistic guarantees on the deviations of all sample means (e.g., sample frequencies, or other measure of interestingness, of all patterns) from their expectations (the exact interestingness of the patterns in the dataset).

The second meaning is *sample* as a *sample from an unknown data generating distribution*: the whole dataset is seen as a collection of samples from an unknown distribution, and the goal of mining patterns from the available dataset is to gain approximate information (or better, discover knowledge) about the distribution. This area is known as *Statistically-sound Pattern Discovery* (Hämäläinen and Webb, 2019), and there are many different flavours of it, from Significant Pattern Mining (Terada et al., 2013a) from transactional datasets (Pellegrina et al., 2019c; Kirsch et al., 2012), sequences (Tonon and Vandin, 2019), or graphs (Sugiyama et al., 2015), to True Frequent Itemset Mining (Riondato and Vandin, 2014), to, at least in part, Contrast Pattern Mining (Bay and Pazzani, 2001). Many works in this area also use concepts from statistical learning theory such as empirical VC dimension (Riondato and Vandin, 2014) or Rademacher averages (Pellegrina et al., 2019c), because, once again, these concepts allow to get very sharp bounds on the maximum difference between the observed interestingness on the sample and the unknown interestingness according to

the distribution.

The two meanings of "sample" are really two sides of the same coin, because also in the first case the goal is to approximate an unknown distribution from a sample, thus falling back into the second case. Despite this similarity, previous contributions have been extremely point-of-view-specific and pattern-specific. In part, these limitations are due to the techniques used to study the trade-off between sample size and quality of the approximation obtained from the sample. Our work instead proposes a *unifying solution* for mining approximate collections of patterns from samples, while giving guarantees on the quality of the approximation: our proposed method can easily be adapted to approximate collections of frequent itemsets, frequent sequences, true frequent patterns, significant patterns, and many other tasks, even outside of pattern mining.

At the core of our approach is the *n-Samples Monte-Carlo (Empirical) Rademacher Average (n-MCERA)* (Bartlett and Mendelson, 2002) (see (6.4)), which has the flexibility and the power needed to achieve our goals, as it gives much sharper bounds to the deviation than other approaches. The challenge in using the n-MCERA, like other quantities from statistical learning theory, is how to compute it efficiently.

Contributions We present MCRapper, an algorithm for the fast computation of the n-MCERA of families of functions with a poset structure, which often arise in pattern mining tasks (Section 6.3.1).

- MCRapper is the first algorithm to compute the n-MCERA efficiently. It achieves this goal by using sharp upper bounds to the discrepancy of each function in the family (Section 6.4.1) to quickly prune large parts of the function search space during the exploration necessary to compute the n-MCERA, in a branch-and-bound fashion. We also develop a novel sharper upper bound to the supremum deviation using the 1-MCERA (Theorem 6.4.6). It holds for any family of functions, and is of independent interest.

- To showcase the practical strength of MCRapper, we develop TFP-R (Section 6.5), a novel algorithm for the extraction of the True Frequent Patterns (TFP) (Riondato and Vandin, 2014). TFP-R gives probabilistic guarantees on the quality of its output: with probability at least $1 - \delta$ (over the choice of the sample and the randomness used in the algorithm), for user-supplied $\delta \in (0, 1)$, the output is guaranteed to not contain any false positives. That is, TFP-R controls the Family-Wise Error Rate (FWER) at level δ while achieving high statistical power, thanks to the use of the n-MCERA and of novel applications of *variance-aware* tail bounds (Theorem 6.3.2). We also discuss other applications of MCRapper, to remark on its flexibility as a general-purpose algorithm.

- We conduct an extensive experimental evaluation of MCRapper and TFP-R on real datasets (Section 6.6), and compare their performance with that of state-

of-the-art algorithms for their respective tasks. MCRapper, thanks to the n-MCERA, computes much sharper (i.e, lower) upper bounds to the supremum deviation than algorithms using the looser Massart's lemma (Shalev-Shwartz and Ben-David, 2014, Lemma 26.8). TFP-R extracts many more TFPs (i.e., has higher statistical power) than existing algorithms with the same guarantees.

6.2 Related Work

Our work applies to both the "small-random-sample-from-large-dataset" and the "dataset-as-a-sample" settings, so we now discuss the relationship of our work to prior art in both settings. We do not study the important but different task of output sampling in pattern mining (Boley et al., 2011; Dzyuba et al., 2017). We focus on works that use concepts from statistical learning theory: these are the most related to our work, and most often the state of the art in their areas. More details are available in surveys (Riondato and Upfal, 2014; Hämäläinen and Webb, 2019).

The idea of mining a small random sample of a large dataset to speed up the pattern extraction step was proposed for the case of itemsets by Toivonen (1996) shortly after the first algorithm for the task had been introduced. The trade-off between the sample size and the quality of the approximation obtained from the sample has been progressively better characterized (Chakaravarthy et al., 2009; Pietracaprina et al., 2010; Riondato and Upfal, 2014, 2015), with large improvements due to the use of concepts from statistical learning theory. Riondato and Upfal (2014) study the VC-dimension of the itemsets mining task, which results in a worst-case *dataset-dependent* but *sample- and distribution-agnostic characterization* of the trade-off. The major advantage of using Rademacher averages (Koltchinskii and Panchenko, 2000), as we do in MCRapper is that the characterization is now *sample-and-distribution-dependent*, which gives much better upper bounds to the maximum deviation of sample means from their expectations. Rademacher averages were also used by Riondato and Upfal (2015), but they used worst-case upper bounds (based on Massart's lemma (Shalev-Shwartz and Ben-David, 2014, Lemma 26.2)) to the empirical Rademacher average of the task, resulting in excessively large bounds. MCRapper instead computes the *exact* n-MCERA of the family of interest on the observed sample, without having to consider the worst case. For other kinds of patterns, Riondato and Vandin (2018) studied the pseudodimension of subgroups, while Servan-Schreiber et al. (2018a) and Santoro et al. (2020) considered the (empirical) VC-dimension and Rademacher averages for sequential patterns. MCRapper can be applied in all these cases, and obtains better bounds because it uses the *sample-and-distribution-dependent* n-MCERA, rather than a worst case dataset-dependent bound.

Significant pattern mining considers the dataset as a sample from an unknown distribution. Many variants and algorithms are described in the survey by Hämäläinen and Webb (2019). We discuss only the two most related to our work. Riondato and Vandin (2014) introduce the problem of finding the true frequent itemsets, i.e., the

itemsets that are frequent w.r.t. the unknown distribution. They propose a method based on empirical VC-dimension to compute the frequency threshold to use to obtain a collection of true frequent patterns with no false positives (see also Section 6.5). Our algorithm TFP-R uses the n-MCERA, and as we show in Section 6.6, it greatly outperforms the state-of-the-art (a modified version of the algorithm by Riondato and Upfal (2015) for approximate frequent itemsets mining). Pellegrina et al. (2019c) use empirical Rademacher averages in their work for Significant Pattern Mining. As their work uses the bound by Riondato and Upfal (2015), the same comments about the n-MCERA being a superior approach hold.

Our approach for bounding the supremum deviation by computing the n-MCERA with efficient search space exploration techniques is novel, not just in knowledge discovery, as the n-MCERA has received scant attention. De Stefani and Upfal (2019) use it to control the generalization error in a sequential and adaptive setting, but do not discuss efficient computation. We believe that the lack of attention to the n-MCERA can be be explained by the fact that there were no efficient algorithms for it, a gap now filled by MCRapper.

6.3 Preliminaries

We now define the most important concepts and results that we use throughout this work, recalling them from Section 2.1.1. Let $\mathcal{F}$ be a class of real valued functions from a domain $\mathcal{X}$ to the interval $[a, b] \subset \mathbb{R}$. We use c to denote $|b-a|$ and z to denote $\max\{|a|, |b|\}$. In this work, we focus on a specific class of families (see Section 6.3.1). In pattern mining from transactional datasets, $\mathcal{X}$ is the set of all possible transactions (or, e.g., sequences). Let μ be an *unknown* probability distribution over $\mathcal{X}$ and the *sample* $\mathcal{S} = \{s_1, \ldots, s_m\}$ be a bag of m i.i.d. random samples from $\mathcal{X}$ drawn according to μ. We discussed in Section 6.1 how in the pattern mining case, the sample may either be the whole dataset (sampled according to an unknown distribution) or a random sample of a large dataset (more details in Section 6.3.1). For each $f \in \mathcal{F}$, we define its *empirical sample average (or sample mean)* $\mathsf{a}_f(\mathcal{S})$ *on* $\mathcal{S}$ and its *expectation* $\mathbb{E}[f]$ respectively as

$$\mathsf{a}_f(\mathcal{S}) \doteq \frac{1}{m} \sum_{s_i \in \mathcal{S}} f(s_i) \text{ and } \mathbb{E}[f] \doteq \mathbb{E}_\mu \left[\frac{1}{m} \sum_{s_i \in \mathcal{S}} f(s_i) \right] .$$

In the pattern mining case, the sample mean is the observed interestingness of a pattern, e.g., its frequency (but other measures of interestingness can be modeled as above, as discussed for subgroups by Riondato and Vandin (2018)), while the expectation is the unknown exact interestingness that we are interested in approximating, that is, either in the large datasets or w.r.t. the unknown data generating distribution. We are interested in developing tight and fast-to-compute upper bounds to the

supremum deviation (SD) $\mathsf{D}(\mathcal{F}, \mathcal{S})$ *of* $\mathcal{F}$ *on* $\mathcal{S}$ between the empirical sample average and the expectation *simultaneously* for all $f \in \mathcal{F}$, defined as

$$\mathsf{D}(\mathcal{F}, \mathcal{S}) = \sup_{f \in \mathcal{F}} |\mathsf{a}_f(\mathcal{S}) - \mathbb{E}_\mu[f]| \quad . \tag{6.1}$$

The supremum deviation allows to quantify how good the estimates obtained from the samples are. Because μ is unknown, it is not possible to compute $\mathsf{D}(\mathcal{F}, \mathcal{S})$ exactly. We introduce concepts such as Monte-Carlo Rademacher Average and results to compute such bounds in Section 6.3.2, but first we elaborate on the specific class of families that we are interested in.

6.3.1 Poset Families and Patterns

A partially-ordered set, or *poset* is a pair $(A, \preceq)$ where A is a set and $\preceq$ is a binary relation between elements of A that is reflexive, anti-symmetric, and transitive. Examples of posets include the $A = \mathbb{N}$ and the obvious "less-than-or-equal-to" ($\leq$) relation, and the powerset of a set of elements and the "subset-or-equal" ($\subseteq$) relation. For any element $y \in A$, we call an element $w \in A$, $w \neq y$ a *descendant* of y (and call y an *ancestor* of w) if $y \preceq w$. Additionally, if $y \preceq w$ and there is no $q \in A$, $q \neq y$, $q \neq w$ such that $y \preceq q \preceq w$, then we say that w is a *child* of y and that y is a *parent* of w. For example, the set $\{0, 2\}$ is a parent of the set $\{0, 2, 5\}$ and an ancestor of the set $\{0, 1, 2, 7\}$, when considering A to be all possible subsets of integers and the $\subseteq$ relation.

In this work we are interested in posets where A is a family $\mathcal{F}$ of functions as in Section 6.3.2, and the relation $\preceq$ is the following: for any $f, g \in \mathcal{F}$

$$f \preceq g \text{ iff } \begin{cases} f(x) \geq g(x) & \text{for every } x \in \mathcal{X} \text{ s.t. } f(x) \geq 0 \\ f(x) \leq g(x) & \text{for every } x \in \mathcal{X} \text{ s.t. } f(x) < 0 \end{cases} . \tag{6.2}$$

The very general but a bit complicated requirement often collapses to much simpler ones as we discuss below. We aim for generality, as our goal is to develop a unifying approach for many pattern mining tasks, for both meanings of "sample", as discussed in Section 6.1. For now, consider for example that requiring $|f(x)| \geq |g(x)|$ for every $x \in \mathcal{X}$ is a specialization of the above more general requirement. We assume to have access to a blackbox function $\mathsf{children}$ that, given any function $f \in \mathcal{F}$, returns the list of children of f according to $\preceq$, and to a blackbox function $\mathsf{minimals}$ that, given $\mathcal{F}$, returns the *minimal elements w.r.t.* $\preceq$, i.e., all the functions $f \in \mathcal{F}$ without any parents. We refer to families that satisfy these conditions as *poset families*, even if the conditions are more about the relation $\preceq$ than about the family. We now discuss how poset families arise in many pattern mining tasks.

In pattern mining, it is assumed to have a language $\mathfrak{L}$ containing the patterns of interest. For example, in itemsets mining (Agrawal et al., 1993), $\mathfrak{L}$ is the set of

all possible *itemsets*, i.e., all non-empty subsets of an alphabet $\mathcal{I}$ of *items*, while in sequential pattern mining (Agrawal and Srikant, 1995), $\mathfrak{L}$ is the set of sequences, and in subgroup discovery (Klösgen, 1992), $\mathfrak{L}$ is set by the user as the set of patterns of interest. In all these cases, for each pattern $\mathcal{P} \in \mathfrak{L}$, it is possible to define a function $f_{\mathcal{P}}$ from the domain $\mathcal{X}$, which is the set of all possible *transactions*, i.e., elementary components of the dataset or of the sample, to an appropriate co-domain $[a, b]$, such that $f_{\mathcal{P}}(x)$ denotes the "value" of the pattern $\mathcal{P}$ on the transaction x. For example, for itemsets mining, $\mathcal{X}$ is all the subsets of $\mathcal{I}$ and $f_{\mathcal{P}}$ maps $\mathcal{X}$ to $\{0, 1\}$ so that $f_{\mathcal{P}}(x) = 1$ iff $\mathcal{P} \subseteq x$ and 0 otherwise. A consequence of this definition is that $\mathsf{a}_f(\mathcal{S})$ is the *frequency* of $\mathcal{P}$ in $\mathcal{S}$, i.e., the fraction of transaction of $\mathcal{S}$ that contain the pattern $\mathcal{P}$. A more complex (due to the nature of the patterns) but similar definition would hold for sequential patterns. For the case of *high-utility itemset mining* (Fournier-Viger et al., 2019), the value of $f_{\mathcal{P}}(x)$ would be the utility of $\mathcal{P}$ in the transaction x. The family $\mathcal{F}$ is the set of the functions $f_{\mathcal{P}}$ for every pattern $\mathcal{P} \in \mathfrak{L}$. Similar reasoning also applies to patterns on graphs, such as graphlets (Ahmed et al., 2015).

Now that we have defined the set that we are interested in, let's comment on the relation $\preceq$ that, together with the set, forms the poset. In the itemsets case, for any two patterns $\mathcal{P}'$ and $\mathcal{P}'' \in \mathfrak{L}$, i.e., for any two functions $f = f_{\mathcal{P}'}$ and $g = f_{\mathcal{P}''} \in \mathcal{F}$, it holds $f \preceq g$ iff $\mathcal{P}' \subseteq \mathcal{P}''$. For sequences, the *subsequence relation* $\sqsubseteq$ defines $\preceq$ instead. In all pattern mining tasks, the only minimal element of $\mathcal{F}$ w.r.t. $\preceq$ is the empty itemset (or sequence) $\emptyset$. Our assumption to have access to the blackboxes children and minimals is therefore very reasonable, because computing these collections is extremely straightforward in all the pattern mining cases we just mentioned and many others.

6.3.2 Rademacher Averages

Here we present Rademacher averages (Koltchinskii and Panchenko, 2000; Bartlett and Mendelson, 2002) and related results at the core of statistical learning theory (Vapnik, 1998). Our presentation uses the most recent and sharper results, and we also introduce new results (Theorem 6.3.2, and later Theorem 6.4.6) that may be of independent interest. For an introduction to statistical learning theory and more details about Rademacher averages, we refer the interested reader to the textbook by Shalev-Shwartz and Ben-David (2014). In this section we consider a generic family $\mathcal{F}$, not necessarily a poset family.

A key quantity to study the supremum deviation (SD) from (6.1) is the *empirical Rademacher average (ERA)* $\hat{\mathsf{R}}(\mathcal{F}, \mathcal{S})$ *of* $\mathcal{F}$ *on* $\mathcal{S}$ (Koltchinskii and Panchenko, 2000; Bartlett and Mendelson, 2002), defined as follows. Let $\boldsymbol{\sigma} = \langle \sigma_1, \ldots, \sigma_m \rangle$ be a collection of m i.i.d. Rademacher random variables, i.e., each taking value in $\{-1, 1\}$ with

equal probability. The ERA of $\mathcal{F}$ on $\mathcal{S}$ is the quantity

$$\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) \doteq \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i f(s_i)\right] \ . \tag{6.3}$$

Computing the ERA $\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right)$ exactly is often intractable, due to the *expectation* over 2^m possible assignments for $\boldsymbol{\sigma}$, and the need to compute a *supremum* for each of these assignments, which precludes many standard techniques for computing expectations. Bounds to the SD are then obtained through efficiently-computable *upper bounds* to the ERA. Massart's lemma (Shalev-Shwartz and Ben-David, 2014, Lemma 26.2) gives a *deterministic* upper bound to the ERA that is often very loose. Monte-Carlo estimation allows to obtain an often sharper *probabilistic* upper bound to the ERA. For $n \geq 1$, let $\boldsymbol{\sigma} \in \{-1,1\}^{n\times m}$ be a $n \times m$ matrix of i.i.d. Rademacher random variables. The *n-Samples Monte-Carlo Empirical Rademacher Average (n-MCERA)* $\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})$ *of* $\mathcal{F}$ *on* $\mathcal{S}$ *using* $\boldsymbol{\sigma}$ is (Bartlett and Mendelson, 2002)

$$\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) \doteq \frac{1}{n}\sum_{j=1}^{n}\sup_{f\in\mathcal{F}}\frac{1}{m}\sum_{s_i\in\mathcal{S}}\boldsymbol{\sigma}_{j,i}f(s_i) \ . \tag{6.4}$$

The n-MCERA allows to obtain probabilistic upper bounds to the SD as follows (proof in Section 6.7). In Section 6.4.3 we show a novel improved bound for the special case $n = 1$ (Theorem 6.4.6).

Theorem 6.3.1. *Let* $\eta \in (0,1)$*. For ease of notation let*

$$\tilde{\mathsf{R}} \doteq \hat{\mathsf{R}}_m^n\left(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}\right) + 2z\sqrt{\frac{\ln\frac{4}{\eta}}{2nm}} \ . \tag{6.5}$$

With probability at least $1-\eta$ *over the choice of* $\mathcal{S}$ *and* $\boldsymbol{\sigma}$*, it holds*

$$\mathsf{D}(\mathcal{F},\mathcal{S}) \leq 2\tilde{\mathsf{R}} + \frac{\sqrt{c(4m\tilde{\mathsf{R}} + c\ln\frac{4}{\eta})\ln\frac{4}{\eta}}}{m} + \frac{c\ln\frac{4}{\eta}}{m} + c\sqrt{\frac{\ln\frac{4}{\eta}}{2m}} \ . \tag{6.6}$$

Sharper upper bounds to $\mathsf{D}(\mathcal{F},\mathcal{S})$ can be obtained with the n-MCERA when more information about $\mathcal{F}$ is available. The proof is in Section 6.7. We use this result for a specific pattern mining task in Section 6.5.

Theorem 6.3.2. *Let* v *be an upper bound to the variance of* every *function in* $\mathcal{F}$*,*

and let $\eta \in (0,1)$. *Define the following quantities*

$$\rho \doteq \mathsf{R}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + 2z\sqrt{\frac{\ln\frac{4}{\eta}}{2nm}}, \tag{6.7}$$

$$r \doteq \rho + \frac{1}{2m}\left(\sqrt{c\left(4m\rho + c\ln\frac{4}{\eta}\right)\ln\frac{4}{\eta}} + c\ln\frac{4}{\eta}\right),$$

$$\varepsilon \doteq 2r + \sqrt{\frac{2\ln\frac{4}{\eta}\,(v+4cr)}{m}} + \frac{c\ln\frac{4}{\eta}}{3m}\ . \tag{6.8}$$

Then, with probability at least $1-\eta$ *over the choice of* $\mathcal{S}$ *and* $\boldsymbol{\sigma}$, *it holds*

$$\mathsf{D}(\mathcal{F},\mathcal{S}) \le \varepsilon\ .$$

Due to the dependency on z in Theorems 6.3.1 and 6.3.2, it is often convenient to use $\hat{\mathsf{R}}_m^n(\mathcal{F}^{\oplus},\mathcal{S},\boldsymbol{\sigma})$ in place of $\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})$ in the above theorems, where $\mathcal{F}^{\oplus}$ denotes the *range-centralized* family of functions obtained by shifting every function in $\mathcal{F}$ by $-a-\frac{c}{2}$. The results still hold for $\mathsf{D}(\mathcal{F},\mathcal{S})$ because the SD is invariant to shifting, but the bounds to the SD usually improve since the corresponding z for the range-centralized family is smaller.

6.4 MCRapper

We now describe and analyze our algorithm MCRapper to efficiently compute the n-MCERA (see (6.4)) for a family $\mathcal{F}$ with the binary relation $\preceq$ defined in (6.2) and the blackbox functions children and minimals described in Section 6.3.1.

6.4.1 Discrepancy Bounds

For $j \in \{1,\ldots,n\}$, we denote as the *j-discrepancy* $\Delta_j(f)$ *of* $f \in \mathcal{F}$ *on* $\mathcal{S}$ *w.r.t.* $\boldsymbol{\sigma}$ the quantity

$$\Delta_j(f) \doteq \sum_{s_i\in\mathcal{S}} \boldsymbol{\sigma}_{j,i} f(s_i)\ .$$

The j-discrepancy is not an *anti-monotonic function*, in the sense that it does not necessarily hold that $\Delta_j(f) \ge \Delta_j(g)$ for every descendant g of $f \in \mathcal{F}$. Clearly, it holds

$$\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) = \frac{1}{nm}\sum_{j=1}^{n}\sup_{f\in\mathcal{F}}\Delta_j(f)\ . \tag{6.9}$$

A naïve computation of the n-MCERA would require enumerating *all* the functions in $\mathcal{F}$ and computing their j-discrepancies, $1 \le j \le n$, in order to find each of the n suprema. We now present novel easy-to-compute *upper bounds* $\widetilde{\Psi}(f)$ and $\Psi_j(f)$

to $\Delta_j(f)$ such that $\widetilde{\Psi}(f) \geq \Delta_j(g)$ and $\Psi_j(f) \geq \Delta_j(g)$ for every $g \in \mathsf{d}(f)$, where $\mathsf{d}(f)$ denote the set of the *descendants* of f w.r.t. $\preceq$. This key property (which is a generalization of *anti-monotonicity* to posets) allows us to derive efficient algorithms for computing the n-MCERA *exactly* without enumerating *all* the functions in $\mathcal{F}$. Such algorithms take a branch-and-bound approach using the upper bounds to $\Delta_j(f)$ to prune large portions of the search space (see Section 6.4.2).

For every $j \in \{1, \ldots, n\}$ and $i \in \{1, \ldots, m\}$, let

$$\boldsymbol{\sigma}^+_{j,i} \doteq \mathbb{1}(\boldsymbol{\sigma}_{j,i} = 1), \text{ and } \boldsymbol{\sigma}^-_{j,i} \doteq \mathbb{1}(\boldsymbol{\sigma}_{j,i} = -1)$$

and for every $f \in F$ and $x \in \mathcal{X}$, define the functions

$$f^+(x) \doteq f(x)\mathbb{1}(f(x) \geq 0), \text{ and } f^-(x) \doteq f(x)\mathbb{1}(f(x) < 0) \ .$$

It holds $f^+(x) \geq 0$ and $f^-(x) \leq 0$ for every $f \in \mathcal{F}$ and $x \in \mathcal{X}$. For every $j \in \{1, \ldots, n\}$ and $f \in \mathcal{F}$, define

$$\begin{aligned} \widetilde{\Psi}(f) &\doteq \sum_{s_i \in \mathcal{S}} |f(s_i)| && \text{and} \\ \Psi_j(f) &\doteq \sum_{s_i \in \mathcal{S}} \boldsymbol{\sigma}^+_{j,i} f^+(s_i) - \sum_{s_i \in \mathcal{S}} \boldsymbol{\sigma}^-_{j,i} f^-(s_i) && . \end{aligned} \tag{6.10}$$

Computationally, these quantities are extremely straightforward to obtain. Both $\widetilde{\Psi}(f)$ and $\Psi_j(f)$ are upper bounds to $\Delta_j(f)$ and to $\Delta_j(g)$ for all $g \in \mathsf{d}(f)$ (proof in Section 6.7).

Theorem 6.4.1. *For any $f \in \mathcal{F}$ and $j \in \{1, \ldots, n\}$, it holds*

$$\max\{\Delta_j(g) \ : \ g \in \mathsf{d}(f) \cup \{f\}\} \leq \Psi_j(f) \leq \widetilde{\Psi}(f) \ .$$

The bounds we derived in this section are *deterministic*. An interesting direction for future research is how to obtain sharper *probabilistic* bounds.

6.4.2 Algorithms

We now use the discrepancy bounds $\widetilde{\Psi}(\cdot)$ and $\Psi_\cdot(\cdot)$ from Section 6.4.1 in our algorithm MCRapper for computing the exact n-MCERA. As the real problem is usually not to only compute the n-MCERA but to actually compute an upper bound to the SD, our description of MCRapper includes this final step, this also enables fair comparison with existing algorithms that use *deterministic* bounds to the ERA to compute an upper bound to the SD (see also Section 6.6).

MCRapper offers probabilistic guarantees on the quality of the bound it computes (proof deferred to after the presentation).

Theorem 6.4.2. *Let $\delta \in (0,1)$. With probability at least $1-\delta$ over the choice of $\mathcal{S}$ and of $\boldsymbol{\sigma}$, the value ε returned by* MCRapper *is such that* $\mathsf{D}(\mathcal{F},\mathcal{S}) \leq \varepsilon$.

The pseudocode of MCRapper is presented in Algorithm 5. The division in functions is useful for reusing parts of the algorithm in later sections (e.g., Algorithm 7). After having sampled the $n \times m$ matrix of i.i.d. Rademacher random variables (line 1), the algorithm calls the function `getSupDevBound` with appropriate parameters, which in turn calls the function `getNMCERA`, the real heart of the algorithm. This function computes the n-MCERA $\hat{\mathsf{R}}^n_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})$ by exploring and pruning the search space (i.e., $\mathcal{F}$) in according to the order of the elements in the priority queue Q (line 8). One possibility is to explore the space in Breadth-First-Search order (so Q is just a FIFO queue), while another is to use the upper bound $\widetilde{\Psi}(f)$ as the priority, with the top element in the queue being the one with maximum priority among those in the queue. Other orders are possible, but we assume that the order is such that all parents of a function are explored before the function, which is reasonable to ensure maximum pruning, and is satisfied by the two mentioned orders. We assume that the priority queue also has a method `delete(`e`)` to delete an element e in the queue. This requirement could be avoided with some additional book-keeping, but it simplifies the presentation of the algorithm.

The algorithm keeps in the quantities ν_j, $j \in \{1,\dots,n\}$, the currently best available lower bound to the quantity $\sup_{f\in\mathcal{F}} \Delta_j(f)$ (see (6.9)), which initially are all $-zm$ (the lowest possible value of a discrepancy). MCRapper also maintains a dictionary $\mathcal{J}$ (line 10), initially empty, whose keys will be elements of $\mathcal{F}$ and the values are subsets of $\{1,\dots,n\}$. The value associated to a key f in the dictionary is a superset of the set of values $j \in \{1,\dots,n\}$ for which $\widetilde{\Psi}(f) \geq \nu_i$, i.e., for which f or one of its descendants *may* be the function attaining the supremum j-discrepancy among all the functions in $\mathcal{F}$ (see (6.9)). A function and all its descendants are pruned when this set is the empty set. The set of keys of the dictionary $\mathcal{J}$ is, at all times, the set of all and only the functions in $\mathcal{F}$ that have ever been added to Q. The last data structure is the set H (line 11), initially empty, which will contain pruned elements of $\mathcal{F}$, in order to avoid visiting either them or their descendants.

MCRapper populates Q and $\mathcal{J}$ by inserting in them the minimal elements of $\mathcal{F}$ w.r.t. $\preceq$ (line 12), using the set $\{1,\dots,n\}$ as the value for these keys in the dictionary. It then enters a loop that keeps iterating as long as there are elements in Q (line 15). The top element f of Q is extracted at the beginning of each iteration (line 16). A set Y, initially empty, is created to maintain a superset of the set of values $j \in \{1,\dots,n\}$ for which a child of f *may* be the function attaining the supremum j-discrepancy among all the functions in $\mathcal{F}$ (see (6.9)). The algorithm then iterates over the elements $j \in \mathcal{J}[f]$ s.t. $\widetilde{\Psi}(f)$ is greater than ν_j (line 18). The elements for which $\widetilde{\Psi}(f) < v_j$ can be ignored because f and its descendants can not attain the supremum of the j-discrepancy in this case, thanks to Theorem 6.4.1. Computing $\widetilde{\Psi}(f)$ is straightforward and can be done even faster if one keeps a frequent-pattern tree or a similar data structure to avoid having to scan $\mathcal{S}$ all the times, but we do not

Algorithm 5: MCRapper

Input: Poset family $\mathcal{F}$, sample $\mathcal{S}$ of size m, $\delta \in (0,1)$, $n \geq 1$
Output: Upper bound to $\mathsf{D}(\mathcal{F}, \mathcal{S})$ with probability $\geq 1 - \delta$.

$\boldsymbol{\sigma} \leftarrow \texttt{draw}(m,\ n)$
$\varepsilon \leftarrow \texttt{getSupDevBound}(\mathcal{F},\ \mathcal{S},\ \delta,\ \boldsymbol{\sigma})$
return ε
Function `getSupDevBound`$(\mathcal{F},\ \mathcal{S},\ \delta,\ \boldsymbol{\sigma})$**:**
    $\tilde{\mathsf{R}} \leftarrow \texttt{getNMCERA}(\mathcal{F},\ \mathcal{S},\ \boldsymbol{\sigma}) + 2z\sqrt{\frac{\ln(4/\delta)}{2nm}}$
    return *r.h.s. of* (6.6) *using* $\eta = \delta$
Function `getNMCERA`$(\mathcal{F},\ \mathcal{S},\ \boldsymbol{\sigma})$**:**
    $Q \leftarrow$ empty priority queue
    foreach $j \in \{1, \ldots, n\}$ **do** $\nu_j \leftarrow -zm$
    $\mathcal{J} \leftarrow$ empty dictionary from $\mathcal{F}$ to subsets of $\{1, \ldots, n\}$
    $H \leftarrow \emptyset$
    foreach $f \in \texttt{minimals}(\mathcal{F})$ **do**
        $Q.\texttt{push}(f)$
        $\mathcal{J}[f] \leftarrow \{1, \ldots, n\}$
    while Q *is not empty* **do**
        $f \leftarrow Q.\texttt{pop}()$
        $Y \leftarrow \emptyset$
        foreach $j \in \mathcal{J}[f]$ *s.t.* $\tilde{\Psi}(f) \geq \nu_j$ **do**
            if $\Psi_j(f) \geq \nu_j$ **then**
                $\nu_j \leftarrow \max\{\nu_j, \Delta_j(f)\}$
                $Y \leftarrow Y \cup \{j\}$
        foreach $g \in \texttt{children}(f) \setminus H$ **do**
            if $g \in \mathcal{J}$ **then** $N \leftarrow \mathcal{J}[g] \cap Y$ **else** $N \leftarrow Y$
            if $N = \emptyset$ **then**
                $H \leftarrow H \cup \{g\}$
                if $g \in \mathcal{J}$ **then** $Q.\texttt{delete}(g)$
            else
                if $g \notin \mathcal{J}$ **then** $Q.\texttt{push}(g)$
                $\mathcal{J}[g] \leftarrow N$
    return $\frac{1}{nm} \sum_{j=1}^{n} \nu_j$

discuss this case for ease of presentation. For the values j that satisfy the condition on line 18, the algorithm computes $\Delta_j(f)$ and updates ν_j to this value if larger than the current value of ν_j (line 20), to maintain the invariant that ν_j stores the highest value of j-discrepancy seen so far (this invariant, together with the one maintained by the pruning strategy, is at the basis of the correctness of MCRapper). Finally, j is added to the set Y (line 21), as it may still be the case that a descendant of f has j-discrepancy higher than ν_j. The algorithm then iterates over the children of f that have not been pruned, i.e., those not in H (line 22). If the child g is such that there is a key g in $\mathcal{J}$ (because before f we visited another parent of g), then let N be $\mathcal{J}[g] \cap Y$, otherwise, let N be Y. The set N is a superset of the indices j s.t. g may attain the supremum j-discrepancy. Indeed for a value j to have this property, it is *necessary* that $\Psi_j(f) \geq \nu_j$ *for every parent* f of j (where the value of ν_j in this expression is the one that ν_j had when f was visited). If $N = \emptyset$, then g and all its descendants can be pruned, which is achieved by adding g to H (line 25) and removing g from Q if it is a key $\mathcal{J}$ (line 26). When $N \neq \emptyset$, first g is added to Q (with the appropriate priority depending on the ordering of Q) if it did not belong to $\mathcal{J}$ yet (line 28), and then $\mathcal{J}[g]$ is set to N (line 29). This operation completes the current loop iteration starting at line 15.

Once Q is empty, the loop ends and the function `getNMCERA()` returns the sum of the values ν_j divided by $n \cdot m$. The returned value is summed to an appropriate term to obtain $\tilde{\mathsf{R}}$ (line 5), which is used to compute the return value of the function `getSupDevBound()` using (6.6) with $\eta = \delta$ (line 6). This value ε is returned in output by MCRapper when it terminates (line 2).

The following result is at the core of the correctness of MCRapper (proof in Section 6.7.)

Lemma 6.4.3. `getNMCERA(`$\mathcal{F}$, $\mathcal{S}$, $\boldsymbol{\sigma}$`)` *returns the value* $\hat{\mathsf{R}}^n_m(\mathcal{F}, \mathcal{S}, \boldsymbol{\sigma})$.

The proof of Theorem 6.4.2 is then just an application of Theorem 6.4.3 and Theorem 6.3.1 (with $\eta = \delta$), as the value ε returned by MCRapper is computed according to (6.6).

Limiting the exploration of the search space

Despite the very efficient pruning strategy made possible by the upper bounds to the j-discrepancy, MCRapper may still need to explore a large fraction of the search space, with negative impact on the running time. We now present a "hybrid" approach that limits this exploration, while still ensuring the guarantees from Theorem 6.4.2.

Let β be any positive value and define

$$\mathcal{G}(\mathcal{S}, \beta) \doteq \left\{ f \in \mathcal{F} : \frac{1}{m} \sum_{i=1}^{m} (f(s_i))^2 \geq \beta \right\},$$

and $\mathcal{K}(\mathcal{S},\beta) = \mathcal{F} \setminus \mathcal{G}(\mathcal{S},\beta)$. In the case of itemsets mining, $\mathcal{G}(\mathcal{S},\beta)$ would be the set of frequent itemsets w.r.t. $\beta \in [0,1]$.

The following result is a consequence of Hoeffding's inequality and a union bound over $n \cdot |\mathcal{K}(\mathcal{S},\beta)|$ events.

Lemma 6.4.4. *Let $\eta \in (0,1)$. Then, with probability at least $1-\eta$ over the choice of $\boldsymbol{\sigma}$, it holds that* simultaneously for all $j \in \{1,\dots,n\}$,

$$\hat{\mathsf{R}}^1_m(\mathcal{K}(\mathcal{S},\beta),\mathcal{S},\boldsymbol{\sigma}_j) \leq \sqrt{\frac{2\beta \log\left(\frac{n|\mathcal{K}(\mathcal{S},\beta)|}{\eta}\right)}{m}} \ . \tag{6.11}$$

The following is an immediate consequence of the above and the definition of n-MCERA.

Theorem 6.4.5. *Let $\eta \in (0,1)$. Then with probability $\geq 1-\eta$ over the choice of $\boldsymbol{\sigma}$, it holds*

$$\begin{aligned}\hat{\mathsf{R}}^n_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) &= \frac{1}{n}\sum_{j=1}^{n} \max\left\{\hat{\mathsf{R}}^1_m(\mathcal{G}(\mathcal{S},\beta),\mathcal{S},\boldsymbol{\sigma}_j), \hat{\mathsf{R}}^1_m(\mathcal{K}(\mathcal{S},\beta),\mathcal{S},\boldsymbol{\sigma}_j)\right\} \\ &\leq \frac{1}{n}\sum_{j=1}^{n} \max\left\{\hat{\mathsf{R}}^1_m(\mathcal{G}(\mathcal{S},\beta),\mathcal{S},\boldsymbol{\sigma}_j), \sqrt{\frac{2\beta \log\left(\frac{n|\mathcal{K}(\mathcal{S},\beta)|}{\eta}\right)}{m}}\right\} \ .\end{aligned}$$

The result of Theorem 6.4.5 is especially useful in situations when it is possible to compute efficiently reasonable upper bounds on the cardinality of $\mathcal{K}(\mathcal{S},\beta)$, possibly using information from $\mathcal{S}$ (but not $\boldsymbol{\sigma}$). For the case of pattern mining, these bounds are often easy to obtain: e.g., in the case of itemsets, it holds $|\mathcal{K}(\mathcal{S},\beta)| \leq \sum_{s_i \in \mathcal{S}} 2^{|s_i|}$, where $|s_i|$ is the number of items in the transaction s_i. Much better bounds are possible, and in many other cases, but we cannot discuss them here due to space limitations.

Combining the above with MCRapper may lead to a significant speed-up thanks to the fact that MCRapper would be exploring only (a subset of) $\mathcal{G}(\mathcal{S},\beta)$ instead of (a subset of) the entire search space $\mathcal{F}$, at the cost of computing an *upper bound* to $\hat{\mathsf{R}}^n_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}_j)$, rather than its exact value. We study this trade-off, which is governed by the choice of β, experimentally in Section 6.6.3. The correctness follows from Theorems 6.3.1, 6.4.2 and 6.4.5, and an application of the union bound.

We now describe this variant MCRapper-H of MCRapper, presented in Algorithm 6. MCRapper-H accepts in input the same parameters of MCRapper, but also the parameters β and $\gamma < \delta$, which controls the confidence of the probabilistic bound from Theorem 6.4.5. After having drawn $\boldsymbol{\sigma}$, MCRapper-H computes the upper bound to $|\mathcal{K}(\mathcal{S},\beta)|$ (line 3), and calls the function `getNMCERA(`$\mathcal{G}(\mathcal{S},\beta)$`, `$\mathcal{S}$`, `$\boldsymbol{\sigma}$`)` (line 2), slightly modified w.r.t. the one on line 30 of Algorithm 5 so it returns the set of n values $\{\nu_1,\dots,\nu_n\}$ instead of their average. Then, it computes $\tilde{\mathsf{R}}$ using the

r.h.s. of (6.11) and returns the bound to the SD obtained from the r.h.s. of (6.6) with $\eta = \delta - \gamma$.

Algorithm 6: MCRapper-H

Input: Poset family $\mathcal{F}$, sample $\mathcal{S}$ of size m, $\delta \in (0,1)$, $\beta \in [0, z^2]$, $\gamma \in (0, \delta)$
Output: Upper bound to $\mathsf{D}(\mathcal{F}, \mathcal{S})$ with prob. $\geq 1 - \delta$.

1 $\boldsymbol{\sigma} \leftarrow$ `draw(`m, n`)`
2 $\{v_1, \ldots, v_n\} \leftarrow$ `getNMCERA(`$\mathcal{G}(\mathcal{S}, \beta)$, $\mathcal{S}$, $\boldsymbol{\sigma}$`)`
3 $\omega \leftarrow$ upper bound to $|\mathcal{K}(\mathcal{S}, \beta)|$
4 $\tilde{\mathsf{R}} \leftarrow \frac{1}{n} \sum_{j=1}^{n} \max \left\{ \frac{v_j}{m}, \sqrt{\frac{2\beta \log\left(\frac{n\omega}{\gamma}\right)}{m}} \right\} + 2z \sqrt{\frac{\ln\left(\frac{4}{\delta - \gamma}\right)}{2nm}}$
5 **return** *r.h.s. of* (6.6) *using* $\eta = \delta - \gamma$

It is not necessary to choose β a-priori, as long as it is chosen without using any information that depends on $\boldsymbol{\sigma}$. In situations where deciding β a-priori is not simple, one may define instead, for a given value of k set by the user, the quantity β_k defined as

$$\beta_k \doteq \min \{\beta : |\mathcal{G}(\mathcal{S}, \beta)| \leq k\}.$$

When the queue Q (line 8 of Algorithm 5) is sorted by decreasing value of $\sum_{i=1}^{n} (f(s_i))^2$, the value k is the maximum number of nodes the branch-and-bound search in `getNMCERA()` may enumerate. We are investigating more refined bounds than Theorem 6.4.5.

6.4.3 Improved Bound for $n = 1$

For the special case of $n = 1$, it is possible to derive a better bound to the SD than the one presented in Theorem 6.3.1. This result is new and of independent interest because it holds for *any* family $\mathcal{F}$. The proof is in Section 6.7.

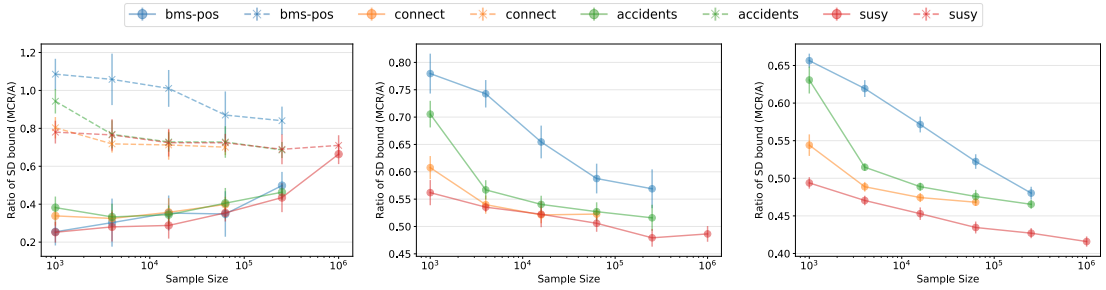
Theorem 6.4.6. *Let the set of functions $\mathcal{F}^{\oplus}$ be composed by functions of $\mathcal{F}$ translated by $-a - \frac{c}{2}$, and let $\eta \in (0,1)$. With probability at least $1 - \eta$ over the choice of $\mathcal{S}$ and σ, it holds that*

$$\mathsf{D}(\mathcal{F}, \mathcal{S}) \leq 2\hat{\mathsf{R}}_m^1 \left(\mathcal{F}^{\oplus}, \mathcal{S}, \boldsymbol{\sigma}\right) + 3c \sqrt{\frac{\ln \frac{2}{\eta}}{2m}}. \tag{6.12}$$

The advantage of (6.12) over (6.6) (with $n = 1$) is in the smaller "tail bounds" terms that arise thanks to a single application of a probabilistic tail bound, rather than three such applications. To use this result in MCRapper, line 2 must be replaced with

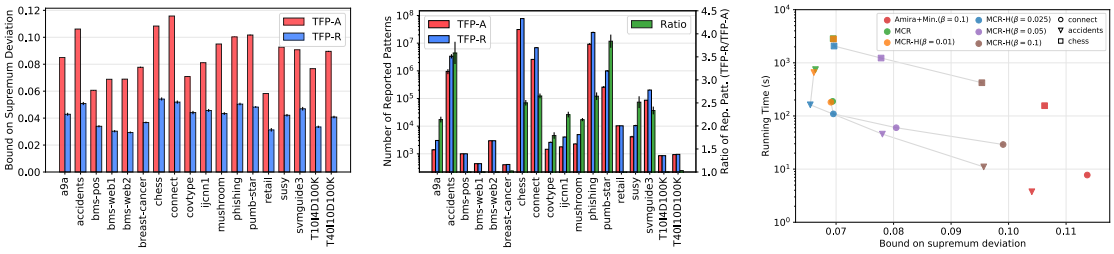
$$\varepsilon \leftarrow \texttt{getNMCERA(}\mathcal{F}^{\oplus}\texttt{, }\mathcal{S}\texttt{, }\boldsymbol{\sigma}\texttt{)} + 3c \sqrt{\frac{\ln \frac{2}{\delta}}{2m}};$$

so the upper bound to the SD is computed according to (6.12). The same guarantees as in Theorem 6.4.2 hold for this modified algorithm.



(a) $n = 1$. (b) $n = 10$. (c) $n = 10^2$.

Figure 6-1: Ratios of the SD Bound obtained by MCRapper ($n \in \{1, 10, 10^2\}$) and amira for the entire $\mathcal{F}$, for 4 of the datasets we analyzed. For $n = 1$, dashed lines use the tail bound from Theorem 6.3.1 instead of the one from Theorem 6.4.6.



(a) (b) (c)

Figure 6-2: (a) Bound on the Supremum Deviation obtained by TFP-R and TFP-A. (b) Number of reported patterns (left y-axis) and ratios (right y-axis) by TFP-R and TFP-A. (c) Running times of MCRapper, MCRapper-H and amira vs corresponding upper bound on SD of the entire $\mathcal{F}$. For MCRapper-H we use different values of β. Each marker shape corresponds to one of the datasets we considered (other 3 shown at the end of the Chapter). For amira we also show the time for mining the TFPs (amira +Min.), with freq. $\geq \beta = 0.1$, as needed after processing the sample.

6.5 Applications

To showcase MCRapper's practical strengths, we now discuss applications to various pattern mining tasks. The value ε computed by MCRapper can be used, for example, to compute, from a random sample $\mathcal{S}$, a high-quality approximation of the

collection of frequent itemsets in a dataset w.r.t. a frequency threshold $\theta \in (0,1)$, by mining the sample at frequency $\theta - \varepsilon$ (Riondato and Upfal, 2014). Also, it can be used in the algorithm by Pellegrina et al. (2019c) to achieve statistical power in Significant Pattern Mining, or in the progressive algorithm by Servan-Schreiber et al. (2018a) to enable even more accurate interactive data exploration. Essentially any of the tasks we mentioned in Sections 6.1 and 6.2 would benefit from the improved bound to the SD computed by MCRapper. To support this claim, we now discuss in depth one specific application.

Mining True Frequent Patterns We now show how to use MCRapper together with sharp variance-aware bounds to the SD (Theorem 6.3.2) for the specific application of identifying the True Frequent Patterns (TFPs) (Riondato and Vandin, 2014). The original work considered the problem only for itemsets, but we solve the problem for a general poset family of functions, thus for many other pattern classes, such as sequences.

The task of TFP mining is, given a pattern language $\mathfrak{L}$ (i.e., a poset family) and a threshold $\theta \in [0,1]$, to output the set

$$\mathsf{TFP}(\theta, \mathfrak{L}) = \{f \in \mathfrak{L} : \mathbb{E}_\mu[f] \geq \theta\} \quad .$$

Computing $\mathsf{TFP}(\theta, \mathfrak{L})$ *exactly* requires to know $\mathbb{E}_\mu[f]$ for all f; since this is almost never the case (and in such case the task is trivial), it is only possible to compute an *approximation* of $\mathsf{TFP}(\theta, \mathfrak{L})$ using information available from a random bag $\mathcal{S}$ of m i.i.d. samples from μ. In this work, mimicking the guarantees given in Significant Pattern Mining (Hämäläinen and Webb, 2019) and in multiple hypothesis testing settings, we are interested in approximations that are a *subset* of $\mathsf{TFP}(\theta, \mathfrak{L})$, i.e., we do not want false *positives* in our approximation, but we accept false *negatives*. A variant that returns a superset of $\mathsf{TFP}(\theta, \mathfrak{L})$ is possible and only requires minimal modifications ot the algorithm. Due to the randomness in the generation of $\mathcal{S}$, no algorithm can guarantee to be able to compute a (non-trivial) subset of $\mathsf{TFP}(\theta, \mathfrak{L})$ from *every* possible $\mathcal{S}$. Thus, one has to accept that there is a probability over the choice of $\mathcal{S}$ *and other random choices made by the algorithm* to obtain a set of patterns that is not a subset of $\mathsf{TFP}(\theta, \mathfrak{L})$. We now present an algorithm TFP-R with the following guarantee (proof in Section 6.7).

Theorem 6.5.1. *Given $\mathfrak{L}$, $\mathcal{S}$, $\theta \in [0,1]$, $\delta \in (0,1)$, and a number $n \geq 1$ of Monte-Carlo trials,* TFP-R *returns a set Y such that*

$$\Pr_{\mathcal{S},\boldsymbol{\sigma}}(Y \subseteq \mathsf{TFP}(\theta, \mathfrak{L})) \geq 1 - \delta,$$

where the probability is over the choice of both $\mathcal{S}$ and the randomness in TFP-R*, i.e., an $n \times m$ matrix of i.i.d. Rademacher variables $\boldsymbol{\sigma}$.*

The intuition for TFP-R is the following. Let $\mathsf{B}^-(\mathsf{TFP}(\theta, \mathfrak{L}))$ be the *negative border* of $\mathsf{TFP}(\theta, \mathfrak{L})$, that is, the set of functions in $\mathfrak{L} \setminus \mathsf{TFP}(\theta, \mathfrak{L})$ such that every parent w.r.t. $\preceq$ of f is in $\mathsf{TFP}(\theta, F)$. If we can compute an $\hat{\varepsilon} \in (0, 1)$ such that, for every $f \in \mathsf{B}^-(\mathsf{TFP}(\theta, \mathfrak{L}))$, it holds $\mathsf{a}_f(\mathcal{S}) \leq \theta + \hat{\varepsilon}$, then we can be sure that any $g \in \mathfrak{L}$ such that $\mathsf{a}_g(\mathcal{S}) > \theta + \hat{\varepsilon}$ belongs to $\mathsf{TFP}(\theta, \mathfrak{L})$. This guarantee will naturally be probabilistic, for the reasons we already discussed. Since $\mathsf{B}^-(\mathsf{TFP}(\theta, \mathfrak{L}))$ is unknown, TFP-R approximates it by *progressively* refining a *superset* $\mathcal{C}$ of it, starting from $\mathfrak{L}$. The correctness of TFP-R is based on the fact that at every point in the execution, it holds $\mathsf{B}^-(\mathsf{TFP}(\theta, \mathfrak{L})) \subseteq \mathcal{C}$, as we show in the proof of Theorem 6.5.1.

Algorithm 7: TFP-R

Input: Poset family $\mathfrak{L}$, sample $\mathcal{S}$ of size m, $\theta \in [0, 1]$, $\delta \in (0, 1)$, $n \geq 1$.
Output: A set Y of patterns

```
1  Y ← ∅
2  σ ← draw(m, n)
3  if θ ≥ 1/2 then v ← 1/4 else v ← θ(1 − θ)
4  C ← 𝔏
5  repeat
6  |  ε̂ ← getSupDevBoundVar(C, S, δ, σ, v)
7  |  C′ ← C
8  |  C ← {f ∈ C′ : a_f(S) < θ + ε̂}
9  |  Y ← Y ∪ (C′ \ C)
10 until C = C′
11 return Y
```

The pseudocode of TFP-R is presented in Algorithm 7. The algorithm first draws the matrix $\boldsymbol{\sigma}$ (line 2), and then computes an upper bound v to the variances of the the frequencies in $\mathsf{B}^-(\mathsf{TFP}(\theta, \mathfrak{L}))$ (line 3). It then initializes, as discussed above, the set $\mathcal{C}$ to $\mathfrak{L}$ (line 4) and enters a loop. At each iteration of the loop, TFP-R calls the function `getSupDevBoundVar` which returns a value $\hat{\varepsilon}$ computed as in (6.8) using $\mathcal{F} = \mathcal{C}$, and $\eta = \delta$. The function `getNMCERA` from Algorithm 5 is used inside of `getSupDevBoundVar` (with parameters $\mathcal{C}$, $\mathcal{S}$, and $\boldsymbol{\sigma}$) to compute the n-MCERA in the value ρ from (6.7). The properties of $\hat{\varepsilon}$ are discussed in the proof for Theorem 6.5.1.

TFP-R uses $\hat{\varepsilon}$ to refine the set $\mathcal{C}$ with the goal of obtaining a better approximation of $\mathsf{B}^-(\mathsf{TFP}(\theta, \mathfrak{L}))$. The set $\mathcal{C}'$ stores the current value of $\mathcal{C}$, and the new value of $\mathcal{C}$ is obtained by keeping all and only the patterns $f \in \mathcal{C}'$ such that $\mathsf{a}_f(\mathcal{S}) < \theta + \hat{\varepsilon}$ (line 8). All the patterns that have been filtered out, i.e., the patterns in $\mathcal{C}' \setminus \mathcal{C}$, or in other words, all the patterns $f \in \mathcal{C}'$ such that $\mathsf{a}_f(\mathcal{S}) \geq \theta + \hat{\varepsilon}$, are added to the output set Y (line 9). TFP-R keeps iterating until the value of $\mathcal{C}$ does not change from the previous iteration (condition on line 10), and finally the set Y is returned in output. While we focused on the a conceptually high-level description of TFP-R, we note that an efficient implementation only requires *one* exploration of $\mathcal{F}$,

such that Y can be provided in output *as $\mathcal{F}$ is explored*, therefore without executing either multiple instances of MCRapper or, at the end of TFP-R, a frequent pattern mining algorithm to compute Y.

6.6 Experiments

In this section we present the results of our experimental evaluation for MCRapper. We compare MCRapper to amira (Riondato and Upfal, 2015), an algorithm that bounds the Supremum Deviation by computing a deterministic upper bound to the ERA with one pass on the random sample. The goal of our experimental evaluation is to compare MCRapper to amira in terms of the upper bound to the SD they compute. We also assess the impact of the difference in the SD bound provided by MCRapper and amira for the application of mining true frequent patterns, by comparing our algorithm TFP-R with TFP-A, a simplified variant of TFP-R that uses amira to compute a bound ε on the SD for all functions in $\mathfrak{L}$, and returns as candidate true frequent patterns the set $\mathcal{G}(\theta + \varepsilon, \mathcal{S})$. It is easy to prove that the output of TFP-A is a subset of true frequent patterns with probability $\geq 1 - \delta$. We also evaluate the running time of MCRapper and of its variant MCRapper-H.

Datasets and implementation We implemented MCRapper and MCRapper-H in `C`, by modifying TopKWY (Pellegrina and Vandin, 2018). Our implementations are available at `https://github.com/VandinLab/MCRapper`. The implementation of amira (Riondato and Upfal, 2015) has been provided by the authors. We test both methods on 18 datasets (see Table 6.1 for their statistics), widely used for the benchmark of frequent itemset mining algorithms. To compare MCRapper to amira in terms of the upper bound to the SD, we draw, from every dataset, random samples of increasing size m; we considered 6 values equally spaced in the logarithmic space in the interval $[10^3, 10^6]$. We only consider values of m smaller than the dataset size $|\mathcal{D}|$. For both algorithms we fix $\delta = 0.1$. For MCRapper we use $n \in \{1, 10, 100\}$.

To compare TFP-R to TFP-A, we analyze synthetic datasets of size $m = 10^4$ obtained by random sampling transactions from each dataset: the true frequency of a pattern corresponds to its frequency in the original dataset, which we use as the ground truth. We use $n = 10$ for TFP-R, and $\delta = 0.1$. We report the results for $\theta = 0.05$ (other values of θ and n produced similar results).

For all experiments and parameters combinations we perform 10 runs (i.e., we create 10 random samples of the same size from the same dataset). In all the figures we report the averages and avg $\pm$ standard deviations of these runs.

6.6.1 Bounds on the SD

Figure 6-1 shows the ratio between the upper bound on the SD obtained by MCRapper and the one obtained by amira for different values of n. The bound provided by

dataset	$\lvert\mathcal{D}\rvert$	$\lvert I\rvert$	avg. trans. len.
svmguide3	1,243	44	21.9
chess	3,196	75	37
breast cancer	7,325	396	11.7
mushroom	8,124	117	22
phishing	11,055	137	30
a9a	32,561	245	13.9
pumb-star	49,046	7,117	50.9
bms-web1	58,136	60,878	3.51
connect	67,557	129	43.5
bms-web2	77,158	330,285	5.6
retail	87,979	16,470	10.8
ijcnn1	91,701	43	13
T10I4D100K	100,000	1,000	10
T40I10D100K	100,000	1,000	40
accidents	340,183	468	34.9
bms-pos	515,420	1,657	6.9
covtype	581,012	108	12.9
susy	5,000,000	190	19

Table 6.1: Datasets statistics. For each dataset, we report the number $|\mathcal{D}|$ of transactions; the number $|\mathcal{I}|$ of items; the average transaction length.

MCRapper is always better (i.e., lower) than the bound provided by amira (e.g., for $n = 100$ the bound from MCRapper is always at least 34% smaller than the bound from amira). For $n = 1$ one can see that the *novel* improved bound from Theorem 6.4.6 should really be preferred over the "standard" one (dashed lines). Similar results hold for all other datasets. These results highlight the effectiveness of MCRapper in providing a much tighter bound to the SD than currently available approaches.

6.6.2 Mining True Frequent Patterns

We compare the *final* SD computed by MCRapper with the one computed by TFP-A. The results are shown in Figure 6-2a. Similarly to what we observed in Section 6.6.1, MCRapper provides much tighter bounds being, in most cases, less than 50% of the bound reported by amira. We then assessed the impact of such difference in the mining of TFP, by comparing the number of patterns reported by TFP-R and by TFP-A. Since for both algorithms the output is a subset of the true frequent patterns with probability $\geq 1-\delta$, reporting a higher number of patterns corresponds to identifying more true frequent patterns, i.e., to higher power. Figure 6-2b shows the number of patterns reported by TFP-R and by TFP-A (left y-axis) and the ratio between such quantities (right y-axis). The SD bound from MCRapper is always lower than the SD bound from amira, so TFP-R always reports at least as many patterns as TFP-A, and for 10 out of 18 datasets, it reports at least *twice* as many patterns as TFP-A. These results show that the SD bound computed by TFP-R provides a great improvement in terms of power for mining TFPs w.r.t. current state-of-the-art methods for SD bound computation.

6.6.3 Running time

For these experiments we take 10 random samples of size 10^4 of the 6 most demanding datasets (`accidents`, `chess`, `connect`, `phishing`, `pumb-star`, `susy`; for the other datasets MCRapper takes much less time than the ones shown) and use the hybrid approach MCRapper-H (Line 30) with different values of β (and $n = 1$, which gives a good trade-off between the bounds and the running time, $\gamma = 0.01$, $\delta = 0.1$). We naïvely upper bound $|\mathcal{K}(\mathcal{S}, \beta)|$ with $\sum_{s_i \in \mathcal{S}} 2^{|s_i|}$, where $|s_i|$ is the length of the transaction s_i, a *very loose* bound that could be improved using more information from $\mathcal{S}$. Figures 6-2c and 6-3 show the running time of MCRapper and amira vs. the obtained upper bound on the SD (different colors correspond to different values of β). With amira one can quickly obtain a fairly loose bound on the SD, by using MCRapper and MCRapper-H one can trade-off the running time for smaller bounds on the SD.

6.7 Proofs and Reproducibility

Missing Proofs

Before proving our proofs, we state some results that will be useful.

Theorem 6.7.1 (Symmetrization inequality (Koltchinskii and Panchenko, 2000))**.** *For any family $\mathcal{F}$ it holds*

$$\mathbb{E}_{\mathcal{S}}\left[\sup_{f\in\mathcal{F}}\left(\mathsf{a}_f\left(\mathcal{S}\right)-\mathbb{E}_\mu[f]\right)-2\hat{\mathsf{R}}(\mathcal{F},\mathcal{S})\right]\leq 0.$$

Theorem 6.7.2 ((Bousquet, 2002), Thm. 2.2)**.** *Let $Z=\sup_{f\in\mathcal{F}}\left(\mathsf{a}_f\left(\mathcal{S}\right)-\mathbb{E}_\mu[f]\right)$. Let $\eta\in(0,1)$. Then, with probability at least $1-\eta$ over the choice of $\mathcal{S}$, it holds*

$$Z\leq\mathbb{E}_\mu\left[Z\right]+\sqrt{\frac{2\ln\frac{1}{\eta}\left(v+2c\mathbb{E}_\mu[Z]\right)}{m}}+\frac{c\ln\frac{1}{\eta}}{3m}\quad.\tag{6.13}$$

Proof of Theorem 6.3.2. Consider the following events

$$\begin{aligned}\mathsf{E}_1&\doteq\rho\geq\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right),\\ \mathsf{E}_2&\doteq E_\mu[\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right)]\leq\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right)\\&\qquad+\frac{1}{2m}\left(\sqrt{c\left(4m\rho+c\ln\frac{4}{\delta}\right)\ln\frac{4}{\delta}}+c\ln\frac{4}{\delta}\right)\quad.\end{aligned}$$

From Theorem 6.7.4, we know that E_1 holds with probability at least $1-\frac{\delta}{4}$ over the choice of $\mathcal{S}$ and $\boldsymbol{\sigma}$. E_2 is guaranteed to with probability at least $1-\frac{\delta}{4}$ over the choice of $\mathcal{S}$ (Oneto et al., 2013, (generalization of) Thm. 3.11). Define the event E_3 as the event in (6.13) for $\eta=\frac{\delta}{4}$ and the event E_4 as the event in (6.13) for $\eta=\frac{\delta}{4}$ and for $\mathcal{F}=-\mathcal{F}$. (Bousquet, 2002, Thm. 2.2) tells us that events E_3 and E_4 hold each with probability at least $1-\frac{d}{4}$ over the choice of $\mathcal{S}$. Thus from the union bound we have that the event $\mathsf{E}=\mathsf{E}_1\cap\mathsf{E}_2\cap\mathsf{E}_3$ holds with probability at least $1-\delta$ over the choice of $\mathcal{S}$ and $\boldsymbol{\sigma}$. Assume from now on that the event E holds.

Because E holds, it must be $r\geq\mathbb{E}_\mu[\hat{\mathsf{R}}(\mathcal{F},\mathcal{S})]$. From this result and Theorem 6.7.1 we have that

$$\mathbb{E}_\mu[\sup_{f\in\mathcal{F}}\left(\mathsf{a}_f\left(\mathcal{S}\right)-\mathbb{E}_\mu[f]\right)]\leq 2\mathbb{E}_\mu[\hat{\mathsf{R}}(\mathcal{F},\mathcal{S})]\leq 2r\quad.$$

From here, and again because E, by plugging $2r$ in place of $E[Z]$ into (6.13) (for $\eta=\frac{\delta}{4}$), we obtain that $\sup_{f\in\mathcal{F}}\left(\mathsf{a}_f\left(\mathcal{S}\right)-\mathbb{E}_\mu[f]\right)\leq\varepsilon$. To show that it also holds

$$\sup_{f\in\mathcal{F}}\left(\mathsf{a}_f\left(\mathcal{S}\right)-\mathbb{E}_\mu[f]\right)\leq\varepsilon$$

(which allows us to conclude that $\mathsf{D}(\mathcal{F},\mathcal{S}) \leq \varepsilon$), we repeat the reasoning above for $-\mathcal{F}$ and use the fact that $\hat{\mathsf{R}}(\mathcal{F},\mathcal{S}) = \hat{\mathsf{R}}(-\mathcal{F},\mathcal{S})$, a known property of the ERA, thus

$$\begin{aligned}&\rho \geq \hat{\mathsf{R}}(-\mathcal{F},\mathcal{S}) \text{ and } r \geq E_\mu[\hat{\mathsf{R}}(-\mathcal{F},\mathcal{S})] \text{ and}\\ &\varepsilon \geq \mathsf{D}(-\mathcal{F},\mathcal{S}) = \sup_{f\in\mathcal{F}} \left(\mathsf{a}_f\left(\mathcal{S}\right) - \mathbb{E}_\mu[f]\right) \ .\end{aligned}$$ □

Theorem 6.7.3 (McDiarmid's inequality (McDiarmid, 1989))**.** *Let $\mathcal{Y} \subseteq \mathbb{R}^\ell$, and let $g : \mathcal{Y} \to \mathbb{R}$ be a function such that, for each i, $1 \leq i \leq \ell$, there is a nonnegative constant c_i such that:*

$$\sup_{\substack{x_1,\ldots,x_\ell\\ x_i'\in\mathcal{X}}} |g(x_1,\ldots,x_\ell) - g(x_1,\ldots,x_{i-1},x_i',x_{i+1},\ldots,x_\ell)| \leq c_i \ . \tag{6.14}$$

Let $x_1,\ldots,x_\ell$ be ℓ independent *random variables taking value in $\mathbb{R}^\ell$ such that $\langle x_1,\ldots,x_\ell\rangle \in \mathcal{Y}$. Then it holds*

$$\Pr\left(g(x_1,\ldots,x_\ell) - \mathbb{E}_\mu[g] > t\right) \leq e^{-2t^2/C},$$

where $C = \sum_{i=1}^\ell c_i^2$.

The following result is an application of McDiarmid's inequality to the n-MCERA, with constants $c_i = \frac{2z}{nm}$.

Lemma 6.7.4. *Let $\eta \in (0,1)$. Then, with probability at least $1-\eta$ over the choice of $\boldsymbol{\sigma}$, it holds*

$$\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})\right] \leq \hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + 2z\sqrt{\frac{\ln\frac{1}{\eta}}{2nm}} \ .$$

The following result gives a probabilistic upper bound to the supremum deviation using the RA and the ERA (Oneto et al., 2013, Thm. 3.11).

Theorem 6.7.5. *Let $\eta \in (0,1)$. Then, with probability at least $1-\eta$ over the choice of $\mathcal{S}$, it holds*

$$\mathsf{D}(\mathcal{F},\mathcal{S}) \leq 2\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) + \frac{\sqrt{c\left(4m\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) + c\ln\frac{3}{\eta}\right)\ln\frac{3}{\eta}}}{m} + \frac{c\ln\frac{3}{\eta}}{m} + c\sqrt{\frac{\ln\frac{3}{\eta}}{2m}} \ .^{1} \tag{6.15}$$

Proof of Theorem 6.3.1. Through Theorem 6.7.4 (using η there equal to $\frac{\eta}{4}$), Theorem 6.7.5 (using η there equal to $\frac{3\eta}{4}$), and an application of the union bound. □

[1] Slightly sharper bounds are possible at the expense of an increased complexity of the terms.

Proof of Theorem 6.4.1. It is immediate from the definitions of $\widetilde{\Psi}(f)$ and $\Psi_j(f)$ in (6.10) that $\Psi_j(f) \leq \widetilde{\Psi}(f)$, so we can focus on $\Psi_j(f)$. We start by showing that $\Delta_j(f) \leq \Psi_j(f)$. It holds

$$\begin{aligned}\Delta_j(f) &= \sum_{s_i \in \mathcal{S}} \boldsymbol{\sigma}^+_{j,i} f^+(s_i) - \sum_{s_i \in \mathcal{S}} \boldsymbol{\sigma}^-_{j,i} f^-(s_i) - \sum_{s_i \in \mathcal{S}} \boldsymbol{\sigma}^-_{j,i} f^+(s_i) + \sum_{s_i \in \mathcal{S}} \boldsymbol{\sigma}^+_{j,i} f^-(s_i) \\ &\leq \sum_{s_i \in \mathcal{S}} \boldsymbol{\sigma}^+_{j,i} f^+(s_i) - \sum_{s_i \in \mathcal{S}} \boldsymbol{\sigma}^-_{j,i} f^-(s_i) = \Psi_j(f)\end{aligned}$$

where the inequality comes from the fact that

$$\sum_{s_i \in \mathcal{S}} \boldsymbol{\sigma}^-_{j,i} f^+(s_i) \geq 0 \text{ , and } \sum_{s_i \in \mathcal{S}} \boldsymbol{\sigma}^+_{j,i} f^-(s_i) \leq 0.$$

To prove that $\Delta_j(g) \leq \Psi_j(f)$ for every $g \in \mathsf{d}(f)$ it is sufficient to show that $\Psi_j(g) \leq \Psi_j(f)$ hold for every such g, since we just showed that $\Delta_j(g) \leq \Psi_j(g)$ is true for any $f \in \mathcal{F}$. It holds $f \preceq g$, so from the definition of the relation $\preceq$ in (6.2), we get

$$\begin{aligned}\Psi_j(g) &= \sum_{s_i \in \mathcal{S}} \boldsymbol{\sigma}^+_{j,i} g^+(s_i) - \sum_{s_i \in \mathcal{S}} \boldsymbol{\sigma}^-_{j,i} g^-(s_i) \\ &\leq \sum_{s_i \in \mathcal{S}} \boldsymbol{\sigma}^+_{j,i} f^+(s_i) - \sum_{s_i \in \mathcal{S}} \boldsymbol{\sigma}^-_{j,i} f^-(s_i) = \Psi_j(f)\end{aligned}$$

which completes our proof. □

Proof of Theorem 6.4.3. For $j \in \{1, \ldots, n\}$, let h_j be any of the functions attaining the supremum in $\sup_{f \in f} \Delta_j(f)$. We need to show that the algorithm updates ν_j on line 20 of Algorithm 5 using $\Delta_j(h_j)$ at some point during its execution. We focus on a single j, as the proof is the same for any value of j.

It is evident from the description of the algorithm that ν_j is always only set to values of $\Delta_j(g)$, and since h_j has the maximum of these values, ν_j will be, at any point in the execution of the algorithm less than or equal to $\Delta_j(h_j)$. Let's call this fact F_1. Thus, if the algorithm ever hits line 20 with $f = h_j$, then we can be sure that the value stored in ν_j will be $\Delta_j(h_j)$, and this variable will never take an higher value. From fact F_1 and Theorem 6.4.1 we also have that at any point in time it must be $\nu_j \leq \Psi_j(h_j) \leq \widetilde{\Psi}(h_j)$, so the conditions on lines 19 and 18 are definitively satisfied, so the question is now whether $j \in \mathcal{J}[h_j]$ and whether there is an iteration of the **while** loop of line 15 for which $f = h_j$.

It holds from Theorem 6.4.1 that it must be $\Delta_j(h_j) \leq \Psi_j(g) \leq \widetilde{\Psi}(g)$ for every ancestor g of h_j. From this fact and from fact A then it holds that at any point in time it must hold $\nu_j \Psi_j(g) \leq \widetilde{\Psi}(g)$ for every such ancestor g of h_j. Thus, the value j is always added to the set Y at every iteration of the **while** loop for which f is an ancestor of h_j. Let's call this fact F_2. Thus, as long as no ancestor of h_j is pruned or

h_j itself is pruned, j is guaranteed to be in $\mathcal{J}[h_j]$. But from fact F_2 and from the fact that j belongs to $\mathcal{J}[f]$ for all the ancestors of h_j that are in `minimals(f)` (line 14), then j must belong to the set N computed on line 23 for all ancestors of h_j, thus N is never empty and therefore no ancestor of h_j is ever pruned and neither is f and we are guaranteed that h_j is added to Q on line 28 when the first of its parents is visited. Thus, there is an iteration of the **while** loop that has $f = h_j$, and because of what we discussed above, then it will be the case that $\nu_j = \Delta_j(h_j)$ and our proof is complete. □

Proof of Theorem 6.4.6. For ease of notation, let $\mathcal{G} = \mathcal{F} - a - \frac{c}{2}$. Consider the event

$$\mathsf{E}_1 \doteq \sup_{g \in \mathcal{G}} \left(\mathsf{a}_g\left(\mathcal{S}\right) - \mathbb{E}_\mu[g]\right) \leq 2\hat{\mathsf{R}}^1_m(\mathcal{G}, \mathcal{S}, \boldsymbol{\sigma}) + 3c\sqrt{\frac{\ln \frac{2}{\eta}}{2m}} \ . \tag{6.16}$$

We now show that this event holds with probability at least $1 - \frac{\eta}{2}$ over the choices of $\mathcal{S}$ and $\boldsymbol{\sigma}$, and then we use this fact to obtain the thesis with some additional steps.

Using linearity of expectation and the fact that the n-MCERA is an unbiased estimator for the ERA (i.e., its expectation is the ERA), we can rewrite the symmetrization inequality (Theorem 6.7.1) as

$$\mathbb{E}_{\mathcal{S},\boldsymbol{\sigma}}\left[\sup_{g \in \mathcal{G}} \left(\mathsf{a}_g\left(\mathcal{S}\right) - \mathbb{E}_\mu[g]\right) - 2\hat{\mathsf{R}}^1_m(\mathcal{G}, \mathcal{S}, \boldsymbol{\sigma})\right] \leq 0 \ .$$

The argument of the (outmost) expectation on the l.h.s. can be seen as a function h of the m pairs of r.v.'s $(\boldsymbol{\sigma}_{1,1}, s_1), \ldots, (\boldsymbol{\sigma}_{1,m}, s_m)$. Fix any possible assignment v' of values to these pairs. Consider now a second assignment v'' obtained from v' by changing the value of *any of the pairs* with any other value in $\{-1, 1\} \times \mathcal{X}$. We want to show that it holds $|h(v') - h(v'')| \leq 3\frac{c}{m}$.

We separately handle the SD and the 1-MCERA, as both depend on the values of the assignment of values to the pairs. The SD does not depend on $\boldsymbol{\sigma}_{1,\cdot}$, and in the argument of the supremum, changing any s_j changes a single summand of the empirical mean $\mathsf{a}_f\left(\mathcal{S}\right)$, with maximal change when $f(s_j)$ changes from a to b (or viceversa), thus the SD itself changes by no more than $\frac{c}{m}$.

We now consider the 1-MCERA, and assume that the pair changing value is $(\boldsymbol{\sigma}_{1,j}, s_j)$. Then the only term of the 1-MCERA sum that changes is the j-th term. If only the first component of the pair changes value (i.e., $\boldsymbol{\sigma}_{1,j}$ changes from 1 to -1 or viceversa, i.e., from $\boldsymbol{\sigma}_{1,j}$ to $-\boldsymbol{\sigma}_{1,j}$), then the j-th term in the 1-MCERA sum cannot change by more than c, because it holds $\boldsymbol{\sigma}_{1,j} g(s_j) \in [-\frac{c}{2}, \frac{c}{2}]$, thus $-\boldsymbol{\sigma}_{1,j} g(s_j)$ also belongs to this interval, and it must be $|\boldsymbol{\sigma}_{1,j} g(s_j) - (-\boldsymbol{\sigma}_{1,j} g(s_j))| \leq c$. If only the second component of the pair changes value (i.e., s_j changes value to $\bar{s}_j$), then the j-th term in the 1-MCERA sum cannot change by more than c, because each function $g \in \mathcal{G}$ goes from $\mathcal{X}$ to $[-\frac{c}{2}, \frac{c}{2}]$, and it must be $|\boldsymbol{\sigma}_{1,j} g(s_j) - \boldsymbol{\sigma}_{i,j} g(\bar{s}_j)| \leq c$. Consider now the final case where both $\boldsymbol{\sigma}_{1,j}$ and s_j change value. We have once again

$|\boldsymbol{\sigma}_{1,j} g(s_j) - (-\boldsymbol{\sigma}_{1,j} g(\bar{s}_j))| \leq c$.

By the adding the maximum change in the SD and the maximum change in the 1-MCERA we can conclude that function h satisfies the requirements of McDiarmid's inequality (Theorem 6.7.3) with constants $3\frac{c}{m}$, and obtain that event E_1 from (6.16) holds with probability at least $1 - \frac{\eta}{2}$.

Let now $-\mathcal{G}$ represent the family of functions containing $-g$ for each $g \in \mathcal{G}$. Consider the event

$$\mathsf{E}_2 \doteq \sup_{g \in -\mathcal{G}} \left(\mathsf{a}_g\left(\mathcal{S}\right) - \mathbb{E}_\mu[g]\right) \leq 2\hat{\mathsf{R}}_m^1(-\mathcal{G}, \mathcal{S}, -\boldsymbol{\sigma}) + 3c\sqrt{\frac{\ln \frac{2}{\eta}}{2m}} \ .$$

Following the same steps as for E_1, we have that E_2 holds with probability at least $1 - \frac{\eta}{2}$, as the fact that we are considering $\hat{\mathsf{R}}_m^1(-\mathcal{G}, \mathcal{S}, -\boldsymbol{\sigma})$ rather than $\hat{\mathsf{R}}_m^1(-\mathcal{G}, \mathcal{S}, \boldsymbol{\sigma})$ is not influential.

It is easy to see that $\hat{\mathsf{R}}_m^1(-\mathcal{G}, \mathcal{S}, -\boldsymbol{\sigma}) = \hat{\mathsf{R}}_m^1(\mathcal{G}, \mathcal{S}, \boldsymbol{\sigma})$, and that

$$\sup_{g \in -\mathcal{G}} \left(\mathsf{a}_g\left(\mathcal{S}\right) - \mathbb{E}_\mu[g]\right) = \sup_{g \in \mathcal{G}} \left(\mathbb{E}_\mu[g] - \mathsf{a}_g\left(\mathcal{S}\right)\right) \ .$$

Thus we can rewrite E_2 as

$$\mathsf{E}_2 = \sup_{g \in \mathcal{G}} \left(\mathbb{E}_\mu[g] - \mathsf{a}_g\left(\mathcal{S}\right)\right) \leq 2\hat{\mathsf{R}}_m^1(\mathcal{G}, \mathcal{S}, \boldsymbol{\sigma}) + 2c\sqrt{\frac{\ln \frac{2}{\eta}}{2m}} \ .$$

From the union bound, we have that E_1 and E_2 hold simultaneously with probability at least $1 - \eta$, i.e., the following event holds with probability at least $1 - \eta$

$$\mathsf{D}(\mathcal{G}, \mathcal{S}, \mu) \leq 2\hat{\mathsf{R}}_m^1(\mathcal{G}, \mathcal{S}, \boldsymbol{\sigma}) + 3c\sqrt{\frac{\ln \frac{2}{\eta}}{2m}} \ .$$

The thesis then follows from the fact $\mathsf{D}(\mathcal{F}, \mathcal{S}) = \mathsf{D}(\mathcal{G}, \mathcal{S}, \mu)$. □

Proof of Theorem 6.5.1. For ease of notation, let $\mathcal{G} = \mathsf{B}^-(\mathsf{TFP}(\theta, \mathfrak{L}))$. Let ρ, r, and ε be as in Theorem 6.3.2 for $\eta = \delta$ and $\mathcal{F} = \mathcal{G}$. Theorem 6.3.2 tells us that, with probability at least $1 - \delta$, it holds $\mathsf{D}(\mathcal{G}, \mathcal{S}) \leq \varepsilon$.[2] Assume from now on that that is the case.

We use this fact to show inductively that, at the end of every iteration of the loop of TFP-R (lines 5–10 of Algorithm 7), it holds that $\mathcal{G} \subseteq \mathcal{C}$ and $Y \subseteq \mathsf{TFP}(\theta, \mathfrak{L})$, and therefore the thesis will hold.

Consider the first iteration of the loop. We have $\mathcal{C} = \mathfrak{L} \supseteq \mathcal{G}$. Let $\hat{\rho}$, $\hat{r}$, and $\hat{\varepsilon}$ be the values computed inside the call to the function `getSupDevBoundVar` on line 6 with

[2]We actually only need a value ε such that $\sup_{f \in \mathcal{G}} \left(\mathsf{a}_f\left(\mathcal{S}\right) - \mathbb{E}_\mu[f]\right) < \varepsilon$, but the gain would be minimal and it would make the presentation more complicated.

the parameters mentioned in the description of the algorithm. It holds that $\hat{\rho} \geq \rho$, because the n-MCERA of a superset of a family is not smaller than the n-MCERA of the family. It follows that $\hat{r} \geq r$, which in turn implies that $\hat{\varepsilon} \geq \varepsilon$. Since we assumed that $\mathsf{D}(\mathcal{G}, \mathcal{S}) \leq \varepsilon$, we have $\hat{\varepsilon} \geq \varepsilon \geq \mathsf{D}(\mathcal{G}, \mathcal{S})$. No function $f \in \mathcal{G}$ may then have sample mean $\mathsf{a}_f(\mathcal{S})$ greater than or equal to $\theta + \hat{\varepsilon}$, as every such f has $\mathbb{E}_\mu[f] < \theta$. Call this fact A. A first consequence of A is that, at the end of the iteration, it holds $\mathcal{G} \subseteq \mathcal{C}$. A second consequence of A and of the antimonotonicity property is that *none* of the functions $f \in \mathfrak{L}$ such that $\mathbb{E}_\mu[f] < \theta$ may have $\mathsf{a}_f(\mathcal{S}) \geq \theta + \hat{\varepsilon}$. Equivalently, only functions $f \in \mathfrak{L}$ such that $\mathbb{E}_\mu[f] \geq \theta$, i.e., such that $f \in \mathsf{TFP}(\theta, \mathfrak{L})$, may have $\mathsf{a}_f(\mathcal{S}) \geq \theta + \hat{\varepsilon}$, i.e., $\mathcal{C}' \setminus \mathcal{C} \subseteq \mathsf{TFP}(\theta, \mathfrak{L})$, so $Y \subseteq \mathsf{TFP}(\theta, \mathfrak{L})$ at the end of the first iteration. The base case is complete.

Assume now that $\mathcal{G} \subseteq \mathcal{C}$ and $Y \subseteq \mathsf{TFP}(\theta, \mathfrak{L})$ at the end of all iterations from 1 to i. Following the same reasoning as for the base case, it holds that these facts are true also at the end of iteration $i + 1$ and our proof is complete. □

Reproducibility

We now describe how to reproduce our experimental results. Code and data are available at `https://github.com/VandinLab/MCRapper`.

The code of MCRapper, TFP-R, and amira are in the sub-folders `mcrapper/` and `amira/`. To compile with recent GCC or Clang, use the `make` command inside each sub-folder.

The convenient scripts `run_amira.py` and `run_mcrapper.py` can be used to run the experiments (i.e., run amira, MCRapper, and TFP-R). They accept many input parameters (described using the flag `-h`). You need to specify a dataset and the size of a random sample to create using the flags `-db` and `-sz`. E.g., to process a random sample of 10^3 transactions from the dataset `mushroom` with $n = 100$, run

```
run_mcrapper.py -db mushroom -sz 1000 -j 100
```

and it automatically executes both amira and MCRapper. The command line to process with TFP-R a sample of 10^4 transactions from the dataset `retail` with $n = 10$ and $\theta = 0.05$ is

```
run_mcrapper.py -db retail -sz 10000 -j 10 -tfp 0.05
```

The `run_all_datasets.py` script runs all the instances of MCRapper and amira in parallel, and can be used to reproduce all the experiments described in Section 6.6. The `run_tfp_all_datasets.py` script reproduces the experiments for TFP-R and TFP-A.

All the results are stored in the files `results_mcrapper.csv` and `results_tfp_mcrapper.csv`.

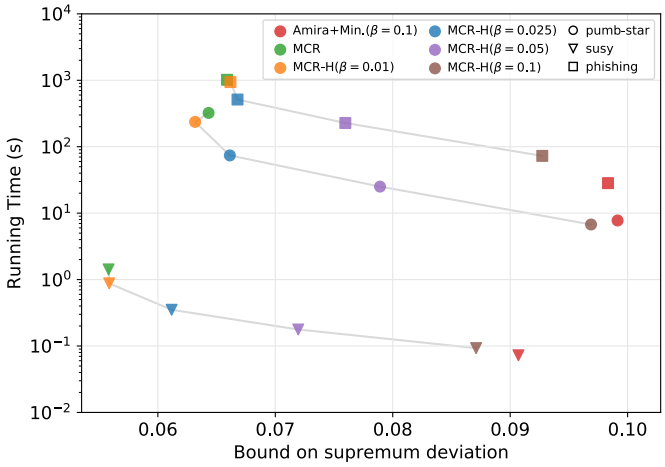


Figure 6-3: Running times of MCRapper, MCRapper-H and amira vs corresponding upper bound on supremum deviation of the entire set of functions $\mathcal{F}$. For MCRapper-H we use different values of β. y-axis in log scale but x axis is linear. Each marker shape corresponds to one of the datasets.

Chapter 7

Sharper Convergence Bounds of Monte Carlo Rademacher Averages through Self-Bounding Functions

7.1 Introduction

In this Chapter we provide new convergence bounds for one of the most interesting notions of data-dependent measure of complexity of sets of functions, the Rademacher Complexity. In particular, we show that it can be estimated in a Monte Carlo way obtaining "faster convergence rates" that depend on characteristic data-dependent quantities of the set of functions.

A potential drawback of Rademacher Averages is that the "global" error that can be obtained may be characterised by the so called "slow" convergence rate of $\mathcal{O}(m^{-1/2})$, where m is the number of analysed samples; while such rate is essentially the best possible when some elements of a set of function $\mathcal{F}$ achieve maximum variance (Boucheron et al., 2005), it may be substantially improved for the other functions, that are often more interesting to the analysis. Therefore, a rich collection of contributions (Koltchinskii and Panchenko, 2000; Massart, 2000; Bousquet et al., 2002; Mendelson, 2002; Bartlett et al., 2005; Koltchinskii, 2006, 2011; Mendelson, 2014) have then focused on providing *local* estimates of the complexity, restricting the estimation to a proper subset of $\mathcal{F}$ that contains only functions with lower variance. In such settings, one would hope to achieve sharper error bounds, with rates between $\mathcal{O}(m^{-1/2})$ and $\mathcal{O}(m^{-1})$.

The slow convergence rate can be attributed to both the "global" computation of Rademacher Averages and from the application of probabilistic concentration inequalities based on the method of bounded differences, that is essentially tight only when there are elements of the set of functions under consideration that achieve maximum variance (Boucheron et al., 2013). Therefore, the study of novel concentration inequalities for the supremum of empirical processes that take advantage of smaller bounds to the variance has been a central focus of research, such as the fundamental contributions of Talagrand (1994, 1995) and many others (Boucheron et al., 2000; Bousquet, 2002; Boucheron et al., 2005, 2013).

The standard approach to bound the Rademacher Complexity is through the application of Massart's Lemma (Massart, 2000). An alternative, often much sharper, approach is to *directly estimate* the Rademacher Averages with the *n-Monte Carlo Empirical Rademacher Average* (n-MCERA) (defined formally in the next Section); this quantity is computed by sampling a finite number of vectors of Rademacher random variables, instead of evaluating its expectation (Bartlett and Mendelson, 2002), and then obtaining a probabilistic upper bound to the Rademacher Complexity with concentration of measure inequalities.

In a recent work, De Stefani and Upfal (2019) used the framework of uniform convergence and Rademacher Complexity to obtain error bounds to empirical averages in an adaptive setting: in their scenario, batch of functions are considered at successive steps, while allowing the choice of the functions to process at every iteration to be based on past information. To quantify the risk of "overfitting", they leverage the n-MCERA, computing it efficiently as functions are processed. Their analysis relates

the n-MCERA to its expectation through Bernstein's inequality and the Central Limit Theorem for martingales.

In other situations, in particular when the size of $\mathcal{F}$ is large, it may be more expensive to compute the n-MCERA, limiting a more widespread practical consideration. We addressed this challenge in Chapter 6 in the context of Significant and Approximate Pattern Mining, deriving a general and practical scheme to compute the n-MCERA by exploiting the combinatorial structure of $\mathcal{F}$ in a branch-and-bound strategy. In all these applications, it is critical to apply sharp concentration results to have tight error rates.

The works we described (De Stefani and Upfal, 2019; Pellegrina et al., 2020a, described in more details in Chapter 6) achieve error bounds that relate the n-MCERA to its expectation, the *Empirical Rademacher Average* (ERA), using concentration inequalities based on the bounded difference property (or, equivalently, assuming maximum variance); for this reason, such error bounds are characterised by the slow convergence rate of $\mathcal{O}((nm)^{-1/2})$, analogous to the worst-case rate of uniform convergence we discussed before. While, in theory, one could use an arbitrary large number n of vectors of Rademacher random variables, and in particular $n = m$ to achieve $\mathcal{O}(m^{-1})$ error rates for estimating the ERA, this would imply the computation of a large number of supremums over $\mathcal{F}$, something impractical in almost all situations.

The question of whether the n-MCERA can be tightly estimated without using an impractically large number of Monte Carlo trials is an unexplored question. In fact, sharp variance-dependent concentration inequalities that relate the n-MCERA to its expectation are not available.

Our contributions. The main goal of this chapter is to provide a positive answer to this question: in Section 7.5 we derive novel concentration bounds for the n-MCERA whose convergence rates depend on characteristic quantities computable from the data, such as the *empirical wimpy-variance* of the set of functions, resulting in a significantly improved trade-off between the guaranteed convergence of the estimate and the number n of required vectors of Rademacher random variables. To do so, we first establish, in Section 7.5.1, *self-bounding* properties of the n-MCERA. Then, we leverage such properties to derive, in Section 7.5.2, novel concentration inequalities for the n-MCERA w.r.t. its expectation, the ERA; such results follow from the sharp exponential concentration inequalities that self-bounding functions satisfy (Boucheron et al., 2000, 2009). Furthermore, in Section 7.5.3 we study the special case of $n = 1$, and prove a novel concentration inequality that directly relates the n-MCERA to the Rademacher Complexity, though the application of Bousquet's inequality (Bousquet, 2002), a central result in Statistical Learning Theory. As the rate of convergence of such bound depends on the unknown *wimpy variance* of the set of functions $\mathcal{F}$, we show that it can be tightly estimated from the available data using its empirical counterpart, the empirical wimpy variance. The guaranteed accuracy of such empirical estimator is proved with the powerful framework of self-bounding functions.

The new bounds we derive in this work are relevant to all methods based on the n-MCERA we introduced before and, given their generality, possibly others. In particular, we believe it would be interesting to fit our results in the framework of Localised Rademacher Averages, and that there are interesting new algorithmic applications of the n-MCERA that may benefit from our results, in particular in problems already tackled with methods based on Rademacher Averages; examples are the analysis of large networks (Riondato and Upfal, 2018; de Lima et al., 2020; Sarpe and Vandin, 2021), rigorous Pattern Mining (Riondato and Upfal, 2015; Santoro et al., 2020), Statistical Hypothesis Testing (Pellegrina et al., 2019c; Li and Barber, 2019), and, potentially, many others.

Another interesting question we explore is whether the maximum difference between empirical averages and their expectation, quantities often denoted by *Supremum Deviations* (SDs), satisfy some form of self-bounding properties. Indeed, after introducing, in Section 7.6, the state-of-the-art variance-dependent bounds to the SDs, in Section 7.7 we show that the SDs are also self-bounding, for appropriate constants that depend on the maximum and minimum expected values of the functions in $\mathcal{F}$; consequently, we derive novel concentration inequalities for the SDs, that may be of independent interest.

7.2 Preliminaries

We denote $\mathcal{F}$ to be a class of real valued functions from a domain $\mathcal{X}$ to the bounded interval $[a, b] \subset \mathbb{R}$, and let $z \doteq \max\{|a|, |b|\}$ and $c \doteq b - a$, with $b > 0 \geq a$, and $c, z > 0$. To simply address non-negativity issues, we assume w.l.o.g. that $\mathcal{F}$ contains a constant function f_0 such that $f_0(x) = 0$, for all $x \in \mathcal{X}$.

Let a sample $\mathcal{S}$ be a bag $\{s_1, \ldots, s_m\}$ of size m, such that $s \in \mathcal{X}, \forall s \in \mathcal{S}$. We assume that each element of $\mathcal{S}$ is drawn i.i.d. from $\mathcal{X}$ according to an unknown probability distribution μ. Our goal is to derive tight bounds on the difference between the average value of f, computed on the sample $\mathcal{S}$, and its expectation $\mathbb{E}[f]$, taken w.r.t. $\mathcal{S}$, that are valid for all functions $f \in \mathcal{F}$. More formally, we define the positive Supremum Deviation (SD) $\mathsf{D}^+(\mathcal{F}, \mathcal{S})$ and the negative supremum deviation $\mathsf{D}^-(\mathcal{F}, \mathcal{S})$ as

$$\mathsf{D}^+(\mathcal{F}, \mathcal{S}) \doteq \sup_{f \in \mathcal{F}} \left\{ \frac{1}{m} \sum_{i=1}^{m} f(s_i) - \mathbb{E}[f] \right\} \quad , \quad \mathsf{D}^-(\mathcal{F}, \mathcal{S}) \doteq \sup_{f \in \mathcal{F}} \left\{ \mathbb{E}[f] - \frac{1}{m} \sum_{i=1}^{m} f(s_i) \right\} \quad .$$

As μ is unknown, it is not possible to directly compute such supremum deviations. However, fundamental results from Statistical Learning Theory allow to obtain probabilistic upper bounds to them, exploiting information obtainable from the data $\mathcal{S}$. We introduce the concepts of *Rademacher Averages*, that will be instrumental to achieve this goal.

First, let $\boldsymbol{\sigma}$ be a $n \times m$ matrix such that each component $\boldsymbol{\sigma}_{i,j}$ of index (i, j) is

either 1 or -1. The *n-Monte Carlo Empirical Rademacher Average* (n-MCERA) $\hat{\mathsf{R}}_m^n(\mathcal{F}, \mathcal{S}, \boldsymbol{\sigma})$ is defined as

$$\hat{\mathsf{R}}_m^n(\mathcal{F}, \mathcal{S}, \boldsymbol{\sigma}) \doteq \frac{1}{n} \sum_{j=1}^{n} \sup_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{\sigma}_{j,i} f(s_i) \ .$$

Denote the *Empirical Rademacher Average* (ERA) $\hat{\mathsf{R}}(\mathcal{F}, \mathcal{S})$ as the expectation of the n-MCERA w.r.t. the assignments of the Rademacher random variables $\boldsymbol{\sigma}$, where each $\boldsymbol{\sigma}_{i,j}$ is 1 or -1 independently and with equal probability:

$$\hat{\mathsf{R}}(\mathcal{F}, \mathcal{S}) \doteq \mathbb{E}_{\boldsymbol{\sigma}} \left[\hat{\mathsf{R}}_m^n(\mathcal{F}, \mathcal{S}, \boldsymbol{\sigma}) \right] \ .$$

Then, denote the *Rademacher Complexity* (RC) $\mathsf{R}(\mathcal{F}, m)$ as the expectation of the ERA over $\mathcal{S}$,

$$\mathsf{R}(\mathcal{F}, m) \doteq \mathbb{E}_{\mathcal{S}} \left[\hat{\mathsf{R}}(\mathcal{F}, \mathcal{S}) \right] \ .$$

The following fundamental result, also known as "Symmetrization lemma", show a precise relationship between the RC and the *expected* supremum deviation (Shalev-Shwartz and Ben-David, 2014; Mitzenmacher and Upfal, 2017).

Lemma 7.2.1.

$$\mathbb{E}_{\mathcal{S}} \left[\mathsf{D}^{+}(\mathcal{F}, \mathcal{S}) \right] \leq 2\mathsf{R}(\mathcal{F}, m) \ ,$$
$$\mathbb{E}_{\mathcal{S}} \left[\mathsf{D}^{-}(\mathcal{F}, \mathcal{S}) \right] \leq 2\mathsf{R}(\mathcal{F}, m) \ .$$

Therefore, upper bounding the RC yields upper bounds on the expected supremum deviations; consequently, one can obtain a probabilistic upper bound on the supremum deviations on the sample $\mathcal{S}$ with the application of concentration inequalities, important tools of probability theory. Most importantly, the RC can be estimated directly on the available data using the n-MCERA. We now define important quantities that will appear in most of our bounds. First, we denote the *wimpy variance* $\sigma_{\mathcal{F}}^2$ of $\mathcal{F}$ as

$$\sigma_{\mathcal{F}}^2 \doteq \sup_{f \in \mathcal{F}} \left\{ \mathbb{E} \left[f^2 \right] \right\} \ .$$

Then, we denote the *empirical wimpy variance* $\hat{\sigma}_{\mathcal{F}}^2(\mathcal{S})$ of $\mathcal{F}$ computed on $\mathcal{S}$ as

$$\hat{\sigma}_{\mathcal{F}}^2(\mathcal{S}) \doteq \frac{1}{m} \sup_{f \in \mathcal{F}} \left\{ \sum_{i=1}^{m} f(s_i)^2 \right\} \ .$$

We also define another quantity of interest $\hat{\nu}_{\mathcal{F}}(\mathcal{S})$, defined as the supremum mean

absolute value of $\mathcal{F}$, computed over $\mathcal{S}$, that is

$$\hat{\nu}_{\mathcal{F}}(\mathcal{S}) \doteq \frac{1}{m} \sup_{f \in \mathcal{F}} \left\{ \sum_{i=1}^{m} |f(s_i)| \right\} \ .$$

In the next Sections we succinctly introduce the most widely used concentration inequalities methods: in Section 7.3.1 we introduce the method of bounded differences; in Section 7.3.2 we present the definitions and recent results on self-bounding functions. The concept of self-bounding functions, as we will discuss later, are essential to prove our novel bounds. We remand for a more exhaustive coverage of the topic to the book of Boucheron et al. (2013).

7.3 Concentration Inequalities

In this Section we introduce two of the most widely employed methods to prove concentration results for functions of independent random variables.

7.3.1 The Method of Bounded Differences

Let $X = (X_1, \ldots, X_n)$ be a vector of variables X_i, each taking values in a measurable set $\mathcal{X}$ and let $g : \mathcal{X}^n \to \mathbb{R}$ be a measurable function. We now introduce the *bounded difference* property, that is often easy to prove in many settings.

Definition 7.3.1 (Bounded difference property)**.** A function g has the *bounded difference property* if, for each i, $1 \leq i \leq m$, there is a nonnegative constant c_i such that:

$$\sup_{\substack{X_1,\ldots,X_m \\ X_i' \in \mathcal{X}}} |g(X_1, \ldots, X_m) - g(X_1, \ldots, X_{i-1}, X_i', X_{i+1}, \ldots, X_m)| \leq c_i \ . \tag{7.1}$$

A central result is given by the following Theorem, that shows that $g(X)$ is well concentrated around its mean $\mathbb{E}[g(X)]$ (taken w.r.t. X), and that the the rate of convergence depends on the constants c_i of the bounded difference property.

Theorem 7.3.2 (McDiarmid (1989))**.** *Let $g : \mathcal{X}^m \to \mathbb{R}$ be a function with the bounded difference property with constants c_i, for $1 \leq i \leq m$. Let $X_1, \ldots, X_m$ be m* independent *random variables taking value in $\mathcal{X}^m$, and let $Z = g(X)$. Then it holds*

$$\Pr\left(Z \geq \mathbb{E}[Z] + t\right) \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{m} c_i^2}\right) \ .$$

Also, it holds

$$\Pr\left(Z \leq \mathbb{E}[Z] - t\right) \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{m} c_i^2}\right) \ .$$

7.3.2 Self-Bounding Functions

Self-bounding functions are an important class of "well-behaving" functions that enjoy sharp concentration inequalities relating their empirical estimates to their expected values, often much tighter than what obtainable through the bounded-difference method. We report their definitions and remand to Boucheron et al. (2013) a more in-depth exposition of the subject.

Let $X = (X_1, \ldots, X_n)$ be a vector of variables X_i, each taking values in a measurable set $\mathcal{X}$ and let $g : \mathcal{X}^n \to \mathbb{R}$ be a non-negative measurable function. Then denote g_i a function from $\mathcal{X}^{n-1} \to \mathbb{R}$. In the following definition, we introduce (α, β)-self-bounding functions; we note that, in some contexts (i.e., in Chapter 6.11 of (Boucheron et al., 2013)), they may also be denoted by *strongly* (α, β)-self-bounding functions.

Definition 7.3.3 ((α, β)-self-bounding function)**.** A function g is a (α, β)-*self-bounding function* if, for all $X \in \mathcal{X}^n$,

$$0 \leq g(X) - g_i(X^{(i)}) \leq 1 \ ,$$

and

$$\sum_{i=1}^{n} \Big(g(X) - g_i(X^{(i)}) \Big) \leq \alpha g(X) + \beta \ ,$$

where $X^{(i)} = (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n) \in \mathcal{X}^{n-1}$ is obtained by dropping the i-th component of X.

An often convenient choice of g_i to prove that g is self-bounding is

$$g_i(X^{(i)}) \doteq \inf_{X_i' \in \mathcal{X}} g\left(X_1, \ldots, X_{i-1}, X_i', X_{i+1}, \ldots, X_n\right) \ .$$

We now introduce *weakly* (α, β)-*self-bounding function.*

Definition 7.3.4 (Weakly (α, β)-self-bounding function)**.** A function g is *weakly* (α, β)-*self-bounding* if, for all $X \in \mathcal{X}^n$,

$$\sum_{i=1}^{n} \Big(g(X) - g_i(X^{(i)}) \Big)^2 \leq \alpha g(X) + \beta \ .$$

Note that a (α, β)-self-bounding function is also a weakly (α, β)-self-bounding function.

The next Theorem shows that if g is self-bounding, then it is sharply concentrated w.r.t. *its expectation* $\mathbb{E}[g(X)]$ (taken w.r.t. X).

Theorem 7.3.5 (Boucheron et al. (2009))**.** *Let $X = (X_1, \ldots, X_n)$ be a vector of independent random variables, each taking values in a measurable set $\mathcal{X}$ and let g :*

$\mathcal{X}^n \to \mathbb{R}$ *be a non-negative measurable function such that* $Z = g(X)$ *has finite mean* $\mathbb{E}[Z] < +\infty$. *Let* $\alpha, \beta \geq 0$, *and define* $\nu = (3\alpha - 1)/6$. *Denote* $(\nu)_+ = \max\{\nu, 0\}$ *and* $(\nu)_- = \max\{-\nu, 0\}$.

If g *is* (α, β)*-self-bounding, then for all* $t > 0$,

$$\Pr\left(Z \geq \mathbb{E}[Z] + t\right) \leq \exp\left(-\frac{t^2}{2\left(\alpha\mathbb{E}[Z] + \beta + (\nu)_+ t\right)}\right).$$

If g *is weakly* (α, β)*-self-bounding and for all* $i \leq n$, *all* $x \in \mathcal{X}$, $g_i(X^{(i)}) \leq g(x)$, *then for all* $t > 0$,

$$\Pr\left(Z \geq \mathbb{E}[Z] + t\right) \leq \exp\left(-\frac{t^2}{2\left(\alpha\mathbb{E}[Z] + \beta + \alpha t/2\right)}\right).$$

If g *is weakly* (α, β)*-self-bounding and* $0 \leq g(X) - g_i(X^{(i)}) \leq 1$[1] *for each* $i \leq n$ *and* $x \in \mathcal{X}^n$, *then for* $0 < t \leq \mathbb{E}[Z]$,

$$\Pr\left(Z \leq \mathbb{E}[Z] - t\right) \leq \exp\left(-\frac{t^2}{2\left(\alpha\mathbb{E}[Z] + \beta + (\nu)_- t\right)}\right).$$

Moreover, if g *is weakly* $(\alpha, 0)$*-self-bounding with* $0 \leq g(X) - g_i(X^{(i)}) \leq 1$ *for all* $i \leq n$ *and* $X \in \mathcal{X}^n$, *then*

$$\Pr\left(Z \leq \mathbb{E}[Z] - t\right) \leq \exp\left(-\frac{t^2}{2\max\{\alpha, 1\}\,\mathbb{E}[Z]}\right).$$

A stronger result for $(1, 0)$-self-bounding functions can be stated.

Theorem 7.3.6 (Boucheron et al. (2000))**.** *Let* $X = (X_1, \ldots, X_n)$ *be a vector of independent random variables, each taking values in a measurable set* $\mathcal{X}$ *and let* $g : \mathcal{X}^n \to \mathbb{R}$ *be a non-negative and bounded measurable function. Let* $h(x) = (1+x)\ln(1+x) - x$.

If $g(X)$ *is a* $(1, 0)$*-self-bounding function, then, it holds, for* $0 < t \leq \mathbb{E}[Z]$,

$$\Pr\left(\mathbb{E}[Z] \geq Z + t\right) \leq \exp\left(-\mathbb{E}[Z]\, h\left(-\frac{t}{\mathbb{E}[Z]}\right)\right),$$

and, for $t > 0$,

$$\Pr\left(Z \geq \mathbb{E}[Z] + t\right) \leq \exp\left(-\mathbb{E}[Z]\, h\left(\frac{t}{\mathbb{E}[Z]}\right)\right).$$

[1]The additional requirement "$0 \leq g(X) - g_i(X^{(i)})$" is not imposed in Theorem 1 of Boucheron et al. (2009) and in Theorem 6.20 of Boucheron et al. (2013), but is required according to the errata of Boucheron et al. (2013), available at `http://www.econ.upf.edu/~lugosi/errata.pdf`. Our results hold regardless of the additional constraint.

7.4 Standard Probabilistic Bounds

In this Section we report standard bounds to the ERA and the SDs, that are proved using the bounded difference method, and a standard bound for the RC based on the self-bounding property of the ERA.

7.4.1 Standard Probabilistic Bound to the ERA

The following result provides a probabilistic upper bound to the ERA from its estimate given by the n-MCERA; it is obtained through the application of the bounded differences method.

Theorem 7.4.1.

$$\Pr\left(\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) \geq \hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \varepsilon\right) \leq \exp\left(\frac{-nm\varepsilon^2}{2z^2}\right).$$

Proof. It is simple to prove that $\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})$ has the bounded difference property with constants $c_i = 2z(nm)^{-1}$, for all $1 \leq i \leq nm$. Therefore, the bound follows from Theorem 7.3.2. □

7.4.2 Standard Probabilistic Bound to the RC

A known property of the ERA is that it is a self-bounding function (see, for instance, Example 3.12 of Boucheron et al. (2013) and Oneto et al. (2013)). This implies concentration bounds, proved by Boucheron et al. (2000), that are often sharper than the ones obtained through the bounded difference property.

Theorem 7.4.2. *Let, for $x \geq -1$, $h(x) \doteq (1+x)\log(1+x) - x$. For all $0 < \varepsilon \leq \mathsf{R}(\mathcal{F},m)$, it holds*

$$\Pr\left(\mathsf{R}(\mathcal{F},m) \geq \hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) + \varepsilon\right) \leq \exp\left(-\frac{m\mathsf{R}(\mathcal{F},m)}{c} h\left(-\frac{\varepsilon}{\mathsf{R}(\mathcal{F},m)}\right)\right) \leq \exp\left(\frac{-m\varepsilon^2}{2c\mathsf{R}(\mathcal{F},m)}\right). \tag{7.2}$$

Also, with probability $\geq 1-\delta$, it holds

$$\mathsf{R}(\mathcal{F},m) \leq \hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) + \frac{c\ln\left(\frac{1}{\delta}\right)}{m} + \sqrt{\left(\frac{c\ln\left(\frac{1}{\delta}\right)}{m}\right)^2 + \frac{2c\ln\left(\frac{1}{\delta}\right)\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right)}{m}}. \tag{7.3}$$

Proof. Equation (7.2) is a consequence of the self-bounding property of the ERA, and therefore follows from Theorem 7.3.6 (Theorem 2.1 of Boucheron et al. (2000), see also Theorem 6.12 of Boucheron et al. (2013)). Equation (7.3) is analogous to what derived by Theorem 3.11 of Oneto et al. (2013). □

From (7.3) it is clear that, as the ERA $\hat{\mathsf{R}}(\mathcal{F}, \mathcal{S})$ gets smaller, the rate of convergence for estimating the RC $\mathsf{R}(\mathcal{F}, m)$ is between $\mathcal{O}(m^{-1/2})$ and $\mathcal{O}(m^{-1})$, an essential improvement in most cases (Boucheron et al., 2013). This intuitively suggests why tight bounds to the ERA are useful and required to reach faster rates of convergence, something not achievable with the "slow rate" bound of Theorem 7.4.1 (at least, not achievable without impractical large n Monte Carlo trials).

7.4.3 Standard Probabilistic Bounds to the SDs

The following result gives standard bounds to the Supremum Deviations using their bounded difference property.

Theorem 7.4.3. *Let* $Z \doteq \sup_{f\in\mathcal{F}} \left\{ \frac{1}{m}\sum_{j=1}^{m} f(s_j) - \mathbb{E}[f] \right\}$. *Then, it holds*

$$\Pr\left(Z \geq \mathbb{E}\left[Z\right] + \varepsilon\right) \leq \exp\left(-\frac{2m\varepsilon^2}{c^2}\right) \quad . \tag{7.4}$$

The same holds for $Z \doteq \sup_{f\in\mathcal{F}} \left\{ \mathbb{E}[f] - \frac{1}{m}\sum_{j=1}^{m} f(s_j) \right\}$.

Proof. It is simple to show that Z has the bounded difference property with constants $c_i = c/m$, for all $1 \leq i \leq m$. Thus, the bounds follows from Theorem 7.3.2. □

In Section 7.6 we will present a well known result that, when additional information on the variance of the functions of $\mathcal{F}$ is available, achieve much stronger bounds to the SDs, matching the rate of convergence of the ERA discussed in Section 7.4.2.

7.5 New Probabilistic Bounds to the ERA

In this Section we show that a careful application of recent results for self-bounding functions allows to prove novel bounds to the ERA from the n-MCERA, whose convergence rates depend on usually easy-to-compute functions of the elements of $\mathcal{F}$ on the sample $\mathcal{S}$. In Section 7.5.1 we show that the n-MCERA is, in fact, (α, β)-self-bounding and weakly (α', β')-self-bounding for appropriate values of α, β, α', and β'. In Section 7.5.2 we show that this implies novel probabilistic bounds to the ERA. First, define $\hat{z}(\mathcal{S})$, the "empirical" version of z, as

$$\hat{z}(\mathcal{S}) = \sup_{s\in\mathcal{S}, f\in\mathcal{F}} |f(s)| \leq z.$$

7.5.1 Self-bounding Properties of the n-MCERA

In this Section we prove self-bounding properties of the n-MCERA. We demand the proofs to Section 7.8.

The first result states the (α, β)-self-bounding property of the n-MCERA.

Theorem 7.5.1. *Let a $n \times m$ matrix $\boldsymbol{\sigma} \in \{-1,1\}^{n\times m}$, and define the function $g(\boldsymbol{\sigma})$ as*

$$g(\boldsymbol{\sigma}) \doteq nm\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) \ .$$

If $\hat{z}(\mathcal{S}) \leq 1/2$, then $g(\boldsymbol{\sigma})$ is a $(1, nm\hat{\nu}_{\mathcal{F}}(\mathcal{S}))$-self-bounding function.

The second result regards the weakly (α,β)-self-bounding property of the n-MCERA.

Theorem 7.5.2. *Let a $n \times m$ matrix $\boldsymbol{\sigma} \in \{-1,1\}^{n\times m}$, and define the function $g(\boldsymbol{\sigma})$ as*

$$g(\boldsymbol{\sigma}) \doteq nm\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) \ .$$

Then $g(\boldsymbol{\sigma})$ is a weakly $(2\hat{z}(\mathcal{S}), 2nm\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S}))$-self-bounding function.

7.5.2 New Probabilistic Bounds on the ERA

In this Section, we show that the self-bounding properties of the n-MCERA we proved yield sharp exponential concentration bounds that relate the n-MCERA to its expectation, the ERA, with significantly improved convergence rates w.r.t. the standard bound of Theorem 7.4.1. All the proofs can be found in Section 7.8.

The first result is based on the self-bounding property of the n-MCERA we proved in Theorem 7.5.1.

Theorem 7.5.3. *Let $\boldsymbol{\sigma} \in \{-1,1\}^{n\times m}$ be an $n \times m$ matrix of Rademacher random variables, such that $\boldsymbol{\sigma}_{j,i} \in \{-1,1\}$ independently and with equal probability. Then, for all $0 < \varepsilon \leq \hat{\mathsf{R}}(\mathcal{F},\mathcal{S})$,*

$$\Pr\left(\hat{\mathsf{R}}(\mathcal{F},\mathcal{S}) \geq \hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \varepsilon\right) \leq \exp\left(-\frac{nm\varepsilon^2}{4\hat{z}(\mathcal{S})\left(\hat{\mathsf{R}}(\mathcal{F},\mathcal{S}) + \hat{\nu}_{\mathcal{F}}(\mathcal{S})\right)}\right) \ . \quad (7.5)$$

The second result is based on the weakly self-bounding property of the n-MCERA we proved in Theorem 7.5.2.

Theorem 7.5.4. *Let $\boldsymbol{\sigma} \in \{-1,1\}^{n\times m}$ be an $n \times m$ matrix of Rademacher random variables, such that $\boldsymbol{\sigma}_{j,i} \in \{-1,1\}$ independently and with equal probability. Then, for all $0 < \varepsilon \leq \hat{\mathsf{R}}(\mathcal{F},\mathcal{S})$,*

$$\Pr\left(\hat{\mathsf{R}}(\mathcal{F},\mathcal{S}) \geq \hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \varepsilon\right) \leq \exp\left(-\frac{nm\varepsilon^2}{4\left(\hat{z}(\mathcal{S})\hat{\mathsf{R}}(\mathcal{F},\mathcal{S}) + \hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right)}\right) \ . \quad (7.6)$$

We may observe that the denominators of the exponents of (7.5) and (7.6) are not known a priori, but depend on the ERA $\hat{\mathsf{R}}(\mathcal{F},\mathcal{S})$, the quantity we actually want

to bound. We remark that plugging an upper bound to $\hat{\mathsf{R}}(\mathcal{F},\mathcal{S})$ is sufficient for the validity of the results. To this aim, we may simply observe that

$$\hat{\mathsf{R}}(\mathcal{F},\mathcal{S}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})\right] \leq \mathbb{E}_{\boldsymbol{\sigma}}\left[\hat{\nu}_{\mathcal{F}}(\mathcal{S})\right] = \hat{\nu}_{\mathcal{F}}(\mathcal{S}) \quad ,$$

obtaining that the r.h.s. of (7.5) and (7.6) are upper bounded by, respectively,

$$\exp\left(-\frac{nm\varepsilon^2}{8\hat{z}(\mathcal{S})\,\hat{\nu}_{\mathcal{F}}(\mathcal{S})}\right) \quad , \text{ and } \quad \exp\left(-\frac{nm\varepsilon^2}{4\left(\hat{z}(\mathcal{S})\,\hat{\nu}_{\mathcal{F}}(\mathcal{S}) + \hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right)}\right) \quad .$$

We now present alternative bounds that only depend on empirical quantities, that are often sharper than plugging the above upper bound to the ERA.

Theorem 7.5.5. *With probability $\geq 1-\delta$ it holds*

$$\hat{\mathsf{R}}(\mathcal{F},\mathcal{S}) \leq \hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \frac{2\hat{z}(\mathcal{S})\ln\left(\frac{1}{\delta}\right)}{nm} + \sqrt{\left(\frac{2\hat{z}(\mathcal{S})\ln\left(\frac{1}{\delta}\right)}{nm}\right)^2 + \frac{4\hat{z}(\mathcal{S})\left(\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \hat{\nu}_{\mathcal{F}}(\mathcal{S})\right)\ln\left(\frac{1}{\delta}\right)}{nm}} \quad .$$

Also, with probability $\geq 1-\delta$, it holds

$$\hat{\mathsf{R}}(\mathcal{F},\mathcal{S}) \leq \hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \frac{2\hat{z}(\mathcal{S})\ln\left(\frac{1}{\delta}\right)}{nm} + \sqrt{\left(\frac{2\hat{z}(\mathcal{S})\ln\left(\frac{1}{\delta}\right)}{nm}\right)^2 + \frac{4\left(\hat{z}(\mathcal{S})\,\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right)\ln\left(\frac{1}{\delta}\right)}{nm}} \quad .$$

We remark that appropriate lower bounds to the ERA $\hat{\mathsf{R}}(\mathcal{F},\mathcal{S})$ can be similarly derived from the self-bounding properties proved in Section 7.5.1 and the application of Theorem 7.3.5.

By directly comparing the bounds we derived by Theorems 7.5.3 and 7.5.4 with the one given by Theorem 7.4.1, we can conclude that the former will be tighter when at least one of the following is satisfied:

$$\hat{\mathsf{R}}(\mathcal{F},\mathcal{S}) + \hat{\nu}_{\mathcal{F}}(\mathcal{S}) \leq \frac{\hat{z}(\mathcal{S})}{2} \quad , \quad 2\hat{z}(\mathcal{S})\,\hat{\mathsf{R}}(\mathcal{F},\mathcal{S}) + 2\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S}) \leq \hat{z}(\mathcal{S})^2 \quad .$$

As discussed before, since $\hat{\mathsf{R}}(\mathcal{F},\mathcal{S}) \leq \hat{\nu}_{\mathcal{F}}(\mathcal{S})$, a sufficient condition for our results to be sharper is given by

$$\hat{\nu}_{\mathcal{F}}(\mathcal{S}) \leq \frac{\hat{z}(\mathcal{S})}{4} \quad , \quad 2\hat{z}(\mathcal{S})\,\hat{\nu}_{\mathcal{F}}(\mathcal{S}) + 2\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S}) \leq \hat{z}(\mathcal{S})^2 \quad .$$

In particular, our novel results allow to bound the ERA $\hat{\mathsf{R}}(\mathcal{F},\mathcal{S})$ below $\hat{\mathsf{R}}^n_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})+\varepsilon$ with an ε of the order of

$$\mathcal{O}\left(\sqrt{\left(\hat{\mathsf{R}}^n_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})+\hat{\nu}_{\mathcal{F}}(\mathcal{S})\right)/nm}\right) \text{ , or } \mathcal{O}\left(\sqrt{\left(\hat{\mathsf{R}}^n_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})+\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right)/nm}\right) ,$$

matching the rate of convergence of the ERA to the RC given by Theorem 7.4.2, instead of the $\mathcal{O}(\sqrt{1/nm})$ slow rate bound. We also remark that, when we bound $\hat{\mathsf{R}}(\mathcal{F},\mathcal{S})$, both $\hat{\nu}_{\mathcal{F}}(\mathcal{S})$ and $\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})$ are deterministic quantities since the sample $\mathcal{S}$ is fixed; thus, they can be used to select the probabilistic result to apply, as they do not depend on the realisation of any random variable.

7.5.3 New Direct Bound for $n = 1$

An interesting case in applications is when only $n = 1$ vector of m Rademacher random variables is used to compute the n-MCERA. In addition of being faster to compute than $n > 1$, Pellegrina et al. (2020a) show that in this case one may obtain a sharper bound to the SDs with only one and direct application of the bounded difference method, considering pairs composed by Rademacher random variables and samples of $\mathcal{S}$ as i.i.d. random variables form an appropriate joint distribution. We now present a variant of their result, that upper bounds the RC instead of the SDs, that is useful to us to be compared with the novel result we prove with Theorem 7.5.7.

Theorem 7.5.6 (Theorem 4.6, (Pellegrina et al., 2020a)). *It holds*

$$\Pr\left(\mathsf{R}(\mathcal{F},m)\geq\hat{\mathsf{R}}^1_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})+\varepsilon\right)\leq\exp\left(-\frac{m\varepsilon^2}{2z^2}\right) ,$$

thus, with probability $\geq 1-\delta$*, it holds*

$$\mathsf{R}(\mathcal{F},m)\leq\hat{\mathsf{R}}^1_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})+z\sqrt{\frac{2\ln\left(\frac{1}{\delta}\right)}{m}} .$$

They also remark that applying the result to the range centralised set of functions

$$\mathcal{F}^{\oplus}\doteq\left\{g : g(x)\doteq f(x)-a-\frac{c}{2}, f\in\mathcal{F}, x\in\mathcal{X}\right\}$$

is often convenient as it gives the sharpest constants in the bound (as z for $\mathcal{F}^{\oplus}$ is equal to $c/2$).

We now derive an analogous but significantly sharper bound, whose convergence rate depends on the wimpy variance $\sigma^2_{\mathcal{F}}$ of $\mathcal{F}$. Our proof, postponed to Section 7.8, is based on the application of a left tail of Bousquet's inequality.

Theorem 7.5.7. *With probability* $\geq 1-\delta$*, it holds*

$$\mathsf{R}(\mathcal{F},m) \leq \hat{\mathsf{R}}^1_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \sqrt{\frac{2(2z\mathsf{R}(\mathcal{F},m)+\sigma^2_{\mathcal{F}})\ln\left(\frac{1}{\delta}\right)}{m}} + \frac{z\ln\left(\frac{1}{\delta}\right)}{8m} \tag{7.7}$$

$$\leq \hat{\mathsf{R}}^1_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \sqrt{\frac{9}{8}\left(\frac{2z\ln\left(\frac{1}{\delta}\right)}{m}\right)^2 + \frac{2(2z\hat{\mathsf{R}}^1_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})+\sigma^2_{\mathcal{F}})\ln\left(\frac{1}{\delta}\right)}{m}} + \frac{17z\ln\left(\frac{1}{\delta}\right)}{8m} \ . \tag{7.8}$$

We may observe that (7.8) may be sharper than the combined application of (7.6) and (7.2) when $n=1$, since the empirical wimpy variance $\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})$ appears in (7.6) with a factor 4, while the wimpy variance in (7.8) has a factor 2. On the other hand, one should have (or compute on the data) an upper bound to $\sigma^2_{\mathcal{F}}$ to apply the result, while Theorem 7.5.4 only requires to compute its empirical counterpart $\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})$.

Nevertheless, we show that the empirical wimpy variance $\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})$ yields a sharp upper bound to the wimpy variance $\sigma^2_{\mathcal{F}}$. Our analysis again relies on the powerful framework of self-bounding functions.

Theorem 7.5.8. *It holds* $\sigma^2_{\mathcal{F}} \leq \mathbb{E}\left[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right]$*, and, for* $\varepsilon \leq \mathbb{E}\left[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right]$*,*

$$\Pr\left(\mathbb{E}\left[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right] \geq \hat{\sigma}^2_{\mathcal{F}}(\mathcal{S}) + \varepsilon\right) \leq \exp\left(-\frac{m\mathbb{E}\left[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right]}{z^2} h\left(-\frac{\varepsilon}{\mathbb{E}\left[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right]}\right)\right) \tag{7.9}$$

$$\leq \exp\left(-\frac{m\varepsilon^2}{2z^2\mathbb{E}\left[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right]}\right) \ . \tag{7.10}$$

Furthermore, with probability $\geq 1-\delta$*, it holds*

$$\sigma^2_{\mathcal{F}} \leq \hat{\sigma}^2_{\mathcal{F}}(\mathcal{S}) + \frac{z^2\ln\left(\frac{1}{\delta}\right)}{m} + \sqrt{\left(\frac{z^2\ln\left(\frac{1}{\delta}\right)}{m}\right)^2 + \frac{2z^2\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\ln\left(\frac{1}{\delta}\right)}{m}} \ . \tag{7.11}$$

This result shows that the empirical wimpy variance $\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})$ is an accurate empirical estimator of the wimpy variance $\sigma^2_{\mathcal{F}}$; we believe such observation may have interesting applications in establishing "global" fast rates of convergence of the SDs, as shown by Oneto et al. (2016).

7.6 Variance-dependent Probabilistic Bounds to the Supremum Deviations

In this section we state a central result in Statistical Learning Theory, due to Bousquet (2002), the sharpest refinement of a number of improvements of the work of Talagrand (1994) on bounds on the deviation of the suprema of empirical processes. This result can be applied to derive bounds on the supremum deviations that depend on the maximum variance $\tau \doteq \sup_{f\in\mathcal{F}}\{Var(f)\}$ of the functions of $\mathcal{F}$. These bounds are often significantly sharper than the ones obtainable with the bounded differences method if τ is sufficiently smaller than its maximum possible value (equal to $c^2/4$ from Popoviciu (1935) inequality on variances).

We first report the result of Bousquet (in the version stated by Theorem A.1 of Bartlett et al. (2005)).

Theorem 7.6.1 (Theorem 2.3, Bousquet (2002))**.** *Let $d > 0$, X_i be independent random variables distributed according to a probability distribution P, and let $\mathcal{G}$ be a set of functions from $\mathcal{X}$ to $\mathbb{R}$. Assume that all functions $g \in \mathcal{G}$ satisfy $\mathbb{E}[g] = 0$ and $\|g\|_\infty \leq d$. Let $\sigma^2 \geq \sup_{g\in\mathcal{G}} Var\left(g(X_i)\right)$. Then, for any $x \geq 0$,*

$$\Pr\left(Z \geq \mathbb{E}[Z] + x\right) \leq \exp\left(-vh\left(\frac{x}{cv}\right)\right) \quad ,$$

where $Z = \sup_{g\in\mathcal{F}} \sum_{i=1}^n g(X_i)$, $h(x) = (1+x)\log(1+x) - x$ and $v = n\sigma^2 + 2d\mathbb{E}[Z]$.

Variance-dependent bounds to the SDs follow from Theorem 7.6.1.

Theorem 7.6.2. *Let $Z \doteq \sup_{f\in\mathcal{F}}\left\{\frac{1}{m}\sum_{j=1}^m f(s_j) - \mathbb{E}[f]\right\}$, and define $\tau \doteq \sup_{f\in\mathcal{F}}\{Var(f)\}$ and the function $h(x) \doteq (1+x)\ln(1+x) - x$. Then, it holds*

$$\Pr\left(Z \geq \mathbb{E}\left[Z\right] + \varepsilon\right) \leq \exp\left(-m\left(\tau + 2c\mathbb{E}\left[Z\right]\right) h\left(\frac{\varepsilon}{\tau + 2c\mathbb{E}\left[Z\right]}\right)\right) \quad . \tag{7.12}$$

Also, with probability at least $1-\delta$, it holds

$$Z \leq \mathbb{E}\left[Z\right] + \sqrt{\frac{2\ln\left(\frac{1}{\delta}\right)\left(\tau + 2c\mathbb{E}[Z]\right)}{m}} + \frac{c\ln\left(\frac{1}{\delta}\right)}{3m} \quad . \tag{7.13}$$

The same results are valid for $Z \doteq \sup_{f\in\mathcal{F}}\left\{\mathbb{E}[f] - \frac{1}{m}\sum_{j=1}^m f(s_j)\right\}$.

7.7 New Probabilistic Bounds to the Supremum Deviations

Bousquet (2003) shows that Theorem 7.6.1 can be applied to analyze the concentration of the supremum of empirical processes for sets of functions satisfying a sub-

additive property, a variant of the $(1,0)$-self-bounding property with relaxed requirements; in fact, the supremum deviation is sub-additive (see Section 6 and Lemma C.1 of (Bousquet, 2003)), but is not, in general, $(1,0)$-self-bounding. Still, in this Section we show that the supremum deviation is $(1,\beta)$-self-bounding, for appropriate values of β that depend on the maximum and minimum expectations of the elements of $\mathcal{F}$. Consequently, we obtain novel bounds to the supremum deviation by applying concentration results for self-bounding functions, similarly to what we did for the n-MCERA.

We first prove self-bounding properties for the supremum deviations. Define $\eta_{\mathcal{F}}$ and $\gamma_{\mathcal{F}}$ as the distance between the boundaries of the codomains of functions in $\mathcal{F}$ and the boundaries of their expectations, such that

$$\eta_{\mathcal{F}} = \sup_{f\in\mathcal{F}} \mathbb{E}[f] - a \ , \ \ \gamma_{\mathcal{F}} = b - \inf_{f\in\mathcal{F}} \mathbb{E}[f] \ .$$

Theorem 7.7.1. *Assume $c \leq 1$. Let $g(\mathcal{S})$ be*

$$g(\mathcal{S}) \doteq m\mathsf{D}^{+}(\mathcal{F},\mathcal{S}) = \sup_{f\in\mathcal{F}} \left\{ \sum_{j=1}^{m} f(s_j) - m\mathbb{E}\left[f\right] \right\} \ .$$

Then, $g(\mathcal{S})$ is a $(1, m\eta_{\mathcal{F}})$-self-bounding function.

Theorem 7.7.2. *Assume $c \leq 1$. Let $g(\mathcal{S})$ be*

$$g(\mathcal{S}) \doteq m\mathsf{D}^{-}(\mathcal{F},\mathcal{S}) = \sup_{f\in\mathcal{F}} \left\{ m\mathbb{E}\left[f\right] - \sum_{j=1}^{m} f(s_j) \right\} \ .$$

Then, $g(\mathcal{S})$ is a $(1, m\gamma_{\mathcal{F}})$-self-bounding function.

We now apply the concentration inequalities given by Theorem 7.3.5 to obtain novel bounds on the supremum deviations. The first results regards the concentration of $\mathsf{D}^{+}(\mathcal{F},\mathcal{S})$.

Theorem 7.7.3. *Let Z be*

$$Z \doteq \mathsf{D}^{+}(\mathcal{F},\mathcal{S}) = \sup_{f\in\mathcal{F}} \left\{ \frac{1}{m}\sum_{j=1}^{m} f(s_j) - \mathbb{E}\left[f\right] \right\} \ .$$

Then, it holds

$$\Pr\left(Z \geq \mathbb{E}\left[Z\right] + \varepsilon\right) \leq \exp\left(-\frac{m\varepsilon^2}{2c\left(\mathbb{E}\left[Z\right] + \eta_{\mathcal{F}} + \varepsilon/3\right)} \right). \tag{7.14}$$

Consequently, with probability $\geq 1-\delta$,

$$Z \leq \mathbb{E}\left[Z\right] + \sqrt{\left(\frac{c\ln\left(\frac{1}{\delta}\right)}{3m}\right)^2 + \frac{2c\ln\left(\frac{1}{\delta}\right)\left(\mathbb{E}\left[Z\right]+\eta_{\mathcal{F}}\right)}{m}} + \frac{c\ln\left(\frac{1}{\delta}\right)}{3m} \quad . \tag{7.15}$$

An analogous result is valid for $\mathsf{D}^{-}(\mathcal{F},\mathcal{S})$.

Theorem 7.7.4. *Let* Z *be*

$$Z \doteq \mathsf{D}^{-}(\mathcal{F},\mathcal{S}) = \sup_{f\in\mathcal{F}}\left\{\mathbb{E}\left[f\right] - \frac{1}{m}\sum_{j=1}^{m} f(s_j)\right\} \quad .$$

Then, it holds

$$\Pr\left(Z \geq \mathbb{E}\left[Z\right]+\varepsilon\right) \leq \exp\left(-\frac{m\varepsilon^2}{2c\left(\mathbb{E}\left[Z\right]+\gamma_{\mathcal{F}}+\varepsilon/3\right)}\right). \tag{7.16}$$

Consequently, with probability $\geq 1-\delta$,

$$Z \leq \mathbb{E}\left[Z\right] + \sqrt{\left(\frac{c\ln\left(\frac{1}{\delta}\right)}{3m}\right)^2 + \frac{2c\ln\left(\frac{1}{\delta}\right)\left(\mathbb{E}\left[Z\right]+\gamma_{\mathcal{F}}\right)}{m}} + \frac{c\ln\left(\frac{1}{\delta}\right)}{3m} \quad . \tag{7.17}$$

We may observe that the novel bounds we proved are less versatile than the result of Bousquet, as they may give faster convergence rates (w.r.t. the bounded difference method) for only one side of the deviation at a time (i.e., either for $\mathsf{D}^{+}(\mathcal{F},\mathcal{S})$ or $\mathsf{D}^{-}(\mathcal{F},\mathcal{S})$) instead of both simultaneously; this is because, for the same $\mathcal{F}$, $\eta_{\mathcal{F}}$ and $\gamma_{\mathcal{F}}$ cannot be both small. However, we observe that such results may be applicable to properly selected *subsets* of $\mathcal{F}$, in a localized fashion. It is not trivial to directly compare these bounds with Bousquet's, in particular (7.12) as it is implicit. However, we observed that our new bounds are slightly sharper than Bousquet's for some range of the values of the quantities involved in the equations, since some of the constants are more favourable. In particular, we can see that, when $c = 1$, the dependence of the additive error term for $\mathbb{E}[Z]$ of (7.15) (and (7.17)) on $\mathbb{E}[Z]$ is lower than the one in (7.13) by a factor $\sqrt{2}$; therefore, when the squared term dominates the error term, (7.15) (resp. (7.17)) is smaller than (7.13) when $\eta_{\mathcal{F}} \leq \mathbb{E}[Z]+\tau$ (resp., $\gamma_{\mathcal{F}} \leq \mathbb{E}[Z]+\tau$). Therefore, we conclude that the combination of Theorem 7.6.2 and our new results gives opportunities to obtain sharper bounds to the SDs of general families of functions.

These results depend on, respectively, the maximum or minimum expected values of elements of $\mathcal{F}$, while Bousquet's inequality requires an upper bound to their maximum variance; a problem in applications is how to handle the cases where these quantities are not known in advance: one intuitive solution is to estimate them from

the data.

Regarding the maximum variance $\tau = \sup_{f\in\mathcal{F}} Var(f)$, we point out that the bounds $\gamma_{\mathcal{F}}$ and $\eta_{\mathcal{F}}$ to the expectations of $f \in \mathcal{F}$ may be sufficient to handle it; in fact, from Bhatia and Davis (2000), one has that, for all f,

$$Var(f) \leq (b - \mathbb{E}[f])(\mathbb{E}[f] - a) \quad , \tag{7.18}$$

with equality when f has binary codomain $\{a, b\}$. Consequently, we have that

$$\begin{aligned}\tau &\leq \sup\left\{(b-x)(x-a) : x \in \left[\inf_{f\in\mathcal{F}} \mathbb{E}[f] \ , \ \sup_{f\in\mathcal{F}} \mathbb{E}[f]\right]\right\} \\ &\leq \max\left\{\gamma_{\mathcal{F}}\left(c - \gamma_{\mathcal{F}}\right), \eta_{\mathcal{F}}\left(c - \eta_{\mathcal{F}}\right)\right\} \leq \gamma_{\mathcal{F}}\eta_{\mathcal{F}} \ .\end{aligned}$$

Therefore, bounds to $\gamma_{\mathcal{F}}$ and $\eta_{\mathcal{F}}$ are of interest, as they suffice for the application of our results and Bousquet's inequality, and may give particularly good bounds for binary functions. Thus, in the following, we show that it is possible to sharply estimate both $\eta_{\mathcal{F}}$ and $\gamma_{\mathcal{F}}$ from the data, analogously to the empirical estimator for the wimpy variance we proved in Section 7.5.3. The proofs are in Section 7.8.

Define the empirical estimators $\hat{\eta}_{\mathcal{F}}(\mathcal{S})$ and $\hat{\gamma}_{\mathcal{F}}(\mathcal{S})$ of, respectively, $\eta_{\mathcal{F}}$ and $\gamma_{\mathcal{F}}$ as

$$\hat{\eta}_{\mathcal{F}}(\mathcal{S}) = \sup_{f\in\mathcal{F}}\left\{\frac{1}{m}\sum_{i=1}^{m} f(s_i)\right\} - a \ , \ \hat{\gamma}_{\mathcal{F}}(\mathcal{S}) = b - \inf_{f\in\mathcal{F}}\left\{\frac{1}{m}\sum_{i=1}^{m} f(s_i)\right\} \ .$$

Theorem 7.7.5. *It holds $\eta_{\mathcal{F}} \leq \mathbb{E}[\hat{\eta}_{\mathcal{F}}(\mathcal{S})]$, and, for $\varepsilon \leq \mathbb{E}[\hat{\eta}_{\mathcal{F}}(\mathcal{S})]$,*

$$\begin{aligned}\Pr\left(\mathbb{E}[\hat{\eta}_{\mathcal{F}}(\mathcal{S})] \geq \hat{\eta}_{\mathcal{F}}(\mathcal{S}) + \varepsilon\right) &\leq \exp\left(-\frac{m\mathbb{E}[\hat{\eta}_{\mathcal{F}}(\mathcal{S})]}{c} h\left(-\frac{\varepsilon}{\mathbb{E}[\hat{\eta}_{\mathcal{F}}(\mathcal{S})]}\right)\right) \\ &\leq \exp\left(-\frac{m\varepsilon^2}{2c\mathbb{E}[\hat{\eta}_{\mathcal{F}}(\mathcal{S})]}\right) \ .\end{aligned} \tag{7.19}$$

Furthermore, with probability $\geq 1 - \delta$, it holds

$$\eta_{\mathcal{F}} \leq \hat{\eta}_{\mathcal{F}}(\mathcal{S}) + \frac{c\ln\left(\frac{1}{\delta}\right)}{m} + \sqrt{\left(\frac{c\ln\left(\frac{1}{\delta}\right)}{m}\right)^2 + \frac{2c\hat{\eta}_{\mathcal{F}}(\mathcal{S})\ln\left(\frac{1}{\delta}\right)}{m}} \ .$$

Theorem 7.7.6. *It holds $\gamma_{\mathcal{F}} \leq \mathbb{E}[\hat{\gamma}_{\mathcal{F}}(\mathcal{S})]$, and, for $\varepsilon \leq \mathbb{E}[\hat{\gamma}_{\mathcal{F}}(\mathcal{S})]$,*

$$\begin{aligned}\Pr\left(\mathbb{E}[\hat{\gamma}_{\mathcal{F}}(\mathcal{S})] \geq \hat{\gamma}_{\mathcal{F}}(\mathcal{S}) + \varepsilon\right) &\leq \exp\left(-\frac{m\mathbb{E}[\hat{\gamma}_{\mathcal{F}}(\mathcal{S})]}{c} h\left(-\frac{\varepsilon}{\mathbb{E}[\hat{\gamma}_{\mathcal{F}}(\mathcal{S})]}\right)\right) \\ &\leq \exp\left(-\frac{m\varepsilon^2}{2c\mathbb{E}[\hat{\gamma}_{\mathcal{F}}(\mathcal{S})]}\right) \ .\end{aligned}$$

Furthermore, with probability $\geq 1-\delta$*, it holds*

$$\gamma_{\mathcal{F}} \leq \hat{\gamma}_{\mathcal{F}}(\mathcal{S}) + \frac{c\ln\left(\frac{1}{\delta}\right)}{m} + \sqrt{\left(\frac{c\ln\left(\frac{1}{\delta}\right)}{m}\right)^2 + \frac{2c\hat{\gamma}_{\mathcal{F}}(\mathcal{S})\ln\left(\frac{1}{\delta}\right)}{m}} \ .$$

7.8 Proofs

Theorem 7.5.1. *Let a* $n \times m$ *matrix* $\boldsymbol{\sigma} \in \{-1,1\}^{n\times m}$*, and define the function* $g(\boldsymbol{\sigma})$ *as*

$$g(\boldsymbol{\sigma}) \doteq nm\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) \ .$$

If $\hat{z}(\mathcal{S}) \leq 1/2$*, then* $g(\boldsymbol{\sigma})$ *is a* $(1, nm\hat{\nu}_{\mathcal{F}}(\mathcal{S}))$*-self-bounding function.*

Proof. Denote the function $g_{j,i}(\boldsymbol{\sigma})$, for $j \in [1,n]$ and $i \in [1,m]$, as

$$g_{j,i}(\boldsymbol{\sigma}) \doteq \inf_{\boldsymbol{\sigma}'_{j,i}\in\{-1,1\}} \left\{ \sum_{\substack{v=1\\v\neq j}}^{n} \left[\sup_{f\in\mathcal{F}} \sum_{h=1}^{m} \boldsymbol{\sigma}_{v,h} f(s_h) \right] + \sup_{f\in\mathcal{F}} \left\{ \sum_{\substack{h=1\\h\neq i}}^{m} (\boldsymbol{\sigma}_{j,h} f(s_h)) + \boldsymbol{\sigma}'_{j,i} f(s_i) \right\} \right\} .$$

This function correspond to $g(\boldsymbol{\sigma})$ where the element $\boldsymbol{\sigma}_{j,i}$ of coordinates (i,j) of $\boldsymbol{\sigma}$ is replaced by $\boldsymbol{\sigma}'_{j,i} \in \{-1,1\}$; in addition, we take the infimum over $\boldsymbol{\sigma}'_{j,i} \in \{-1,1\}$. We remark that, even if $\boldsymbol{\sigma}$ is the argument of $g_{j,i}$ to simplify notation, $\boldsymbol{\sigma}_{j,i}$ never appears in the definition of $g_{j,i}(\boldsymbol{\sigma})$, as required in the definition of self-bounding functions. To show that $g(\boldsymbol{\sigma})$ is (α,β)-self-bounding, according to the definition, we have to show that, for all $\boldsymbol{\sigma} \in \{-1,1\}^{n\times m}$, the inequalities

$$0 \leq g(\boldsymbol{\sigma}) - g_{j,i}(\boldsymbol{\sigma}) \leq 1 \ ,$$

and

$$\sum_{j=1}^{n}\sum_{i=1}^{m}(g(\boldsymbol{\sigma}) - g_{j,i}(\boldsymbol{\sigma})) \leq \alpha g(\boldsymbol{\sigma}) + \beta \tag{7.20}$$

all hold for some non-negative α and β. First, $g(\boldsymbol{\sigma}) \geq g_{j,i}(\boldsymbol{\sigma})$ follows from writing $g_{j,i}(\boldsymbol{\sigma})$ as

$$\begin{aligned} g_{j,i}(\boldsymbol{\sigma}) = \min \Bigg[& \sum_{\substack{v=1\\v\neq j}}^{n} \left[\sup_{f\in\mathcal{F}} \sum_{h=1}^{m} \boldsymbol{\sigma}_{v,h} f(s_h) \right] + \sup_{f\in\mathcal{F}} \left\{ \sum_{\substack{h=1\\h\neq i}}^{m} (\boldsymbol{\sigma}_{j,h} f(s_h)) - f(s_i) \right\} , \\ & \sum_{\substack{v=1\\v\neq j}}^{n} \left[\sup_{f\in\mathcal{F}} \sum_{h=1}^{m} \boldsymbol{\sigma}_{v,h} f(s_h) \right] + \sup_{f\in\mathcal{F}} \left\{ \sum_{\substack{h=1\\h\neq i}}^{m} (\boldsymbol{\sigma}_{j,h} f(s_h)) + f(s_i) \right\} \Bigg] \ , \end{aligned}$$

and from the observation that one argument of the min is equal to $g(\boldsymbol{\sigma})$, therefore the minimum is either equal to $g(\boldsymbol{\sigma})$ or $< g(\boldsymbol{\sigma})$. We now prove that, if $z \leq 1/2$, $g(\boldsymbol{\sigma}) \leq g_{j,i}(\boldsymbol{\sigma}) + 1$, for all $\boldsymbol{\sigma}$ and for all j and i.

$$
\begin{aligned}
g_{j,i}(\boldsymbol{\sigma}) &= \inf_{\boldsymbol{\sigma}'_{j,i} \in \{-1,1\}} \left\{ \sum_{\substack{v=1 \\ v \neq j}}^{n} \left[\sup_{f \in \mathcal{F}} \sum_{h=1}^{m} \boldsymbol{\sigma}_{v,h} f(s_h) \right] + \sup_{f \in \mathcal{F}} \left\{ \sum_{\substack{h=1 \\ h \neq i}}^{m} (\boldsymbol{\sigma}_{j,h} f(s_h)) + \boldsymbol{\sigma}'_{j,i} f(s_i) \right\} \right\} \\
&= \sum_{\substack{v=1 \\ v \neq j}}^{n} \left[\sup_{f \in \mathcal{F}} \sum_{h=1}^{m} \boldsymbol{\sigma}_{v,h} f(s_h) \right] + \inf_{\boldsymbol{\sigma}'_{j,i} \in \{-1,1\}} \left\{ \sup_{f \in \mathcal{F}} \left\{ \sum_{\substack{h=1 \\ h \neq i}}^{m} (\boldsymbol{\sigma}_{j,h} f(s_h)) + \boldsymbol{\sigma}'_{j,i} f(s_i) \right\} \right\} \\
&\geq \sum_{\substack{v=1 \\ v \neq j}}^{n} \left[\sup_{f \in \mathcal{F}} \sum_{h=1}^{m} \boldsymbol{\sigma}_{v,h} f(s_h) \right] + \sup_{f \in \mathcal{F}} \left\{ \inf_{\boldsymbol{\sigma}'_{j,i} \in \{-1,1\}} \left\{ \sum_{\substack{h=1 \\ h \neq i}}^{m} (\boldsymbol{\sigma}_{j,h} f(s_h)) + \boldsymbol{\sigma}'_{j,i} f(s_i) \right\} \right\} \\
&= \sum_{\substack{v=1 \\ v \neq j}}^{n} \left[\sup_{f \in \mathcal{F}} \sum_{h=1}^{m} \boldsymbol{\sigma}_{v,h} f(s_h) \right] + \sup_{f \in \mathcal{F}} \left\{ \sum_{\substack{h=1 \\ h \neq i}}^{m} (\boldsymbol{\sigma}_{j,h} f(s_h)) + \inf_{\boldsymbol{\sigma}'_{j,i} \in \{-1,1\}} \left\{ \boldsymbol{\sigma}'_{j,i} f(s_i) \right\} \right\}.
\end{aligned}
$$

Let $f_j^\star$ be one of the functions of $\mathcal{F}$ attaining the supremum of $\sup_{f \in \mathcal{F}} \sum_{h=1}^{m} \boldsymbol{\sigma}_{j,h} f(s_h)$. Then we continue

$$
\begin{aligned}
g_{j,i}(\boldsymbol{\sigma}) &\geq \sum_{\substack{v=1 \\ v \neq j}}^{n} \left[\sup_{f \in \mathcal{F}} \sum_{h=1}^{m} \boldsymbol{\sigma}_{v,h} f(s_h) \right] + \sup_{f \in \mathcal{F}} \left\{ \sum_{\substack{h=1 \\ h \neq i}}^{m} (\boldsymbol{\sigma}_{j,h} f(s_h)) + \inf_{\boldsymbol{\sigma}'_{j,i} \in \{-1,1\}} \left\{ \boldsymbol{\sigma}'_{j,i} f(s_i) \right\} \right\} \\
&\geq \sum_{\substack{v=1 \\ v \neq j}}^{n} \left[\sup_{f \in \mathcal{F}} \sum_{h=1}^{m} \boldsymbol{\sigma}_{v,h} f(s_h) \right] + \sum_{\substack{h=1 \\ h \neq i}}^{m} \left(\boldsymbol{\sigma}_{j,h} f_j^\star(s_h) \right) + \inf_{\boldsymbol{\sigma}'_{j,i} \in \{-1,1\}} \left\{ \boldsymbol{\sigma}'_{j,i} f_j^\star(s_i) \right\} \\
&= \sum_{\substack{v=1 \\ v \neq j}}^{n} \left[\sup_{f \in \mathcal{F}} \sum_{h=1}^{m} \boldsymbol{\sigma}_{v,h} f(s_h) \right] + \sum_{\substack{h=1 \\ h \neq i}}^{m} \left(\boldsymbol{\sigma}_{j,h} f_j^\star(s_h) \right) + \boldsymbol{\sigma}_{j,i} f_j^\star(s_i) - \boldsymbol{\sigma}_{j,i} f_j^\star(s_i) \\
&\qquad + \inf_{\boldsymbol{\sigma}'_{j,i} \in \{-1,1\}} \left\{ \boldsymbol{\sigma}'_{j,i} f_j^\star(s_i) \right\} \\
&= g(\boldsymbol{\sigma}) - \boldsymbol{\sigma}_{j,i} f_j^\star(s_i) + \inf_{\boldsymbol{\sigma}'_{j,i} \in \{-1,1\}} \left\{ \boldsymbol{\sigma}'_{j,i} f_j^\star(s_i) \right\} \quad . \qquad (7.21)
\end{aligned}
$$

We first observe that

$$
\inf_{\boldsymbol{\sigma}'_{j,i} \in \{-1,1\}} \left\{ \boldsymbol{\sigma}'_{j,i} f_j^\star(s_i) \right\} = \begin{cases} 0 & , \text{ if } f_j^\star(s_i) = 0 \ , \\ -f_j^\star(s_i) & , \text{ if } f_j^\star(s_i) > 0 \ , \\ f_j^\star(s_i) & , \text{ if } f_j^\star(s_i) < 0 \ , \end{cases}
$$

obtaining

$$\inf_{\boldsymbol{\sigma}'_{j,i}\in\{-1,1\}}\left\{\boldsymbol{\sigma}'_{j,i}f_j^\star(s_i)\right\} = -\left|f_j^\star(s_i)\right| \ .$$

Therefore, we continue from (7.21) as follows:

$$g_{j,i}(\boldsymbol{\sigma}) \geq g(\boldsymbol{\sigma}) - \boldsymbol{\sigma}_{j,i}f_j^\star(s_i) - \left|f_j^\star(s_i)\right| \geq g(\boldsymbol{\sigma}) - 2\hat{z}\left(\mathcal{S}\right) \geq g(\boldsymbol{\sigma}) - 1 \ .$$

We now prove (7.20) for $\alpha = 1$ and $\beta = nm\hat{\nu}_{\mathcal{F}}\left(\mathcal{S}\right)$.

$$\begin{aligned}
&\sum_{j=1}^{n}\sum_{i=1}^{m}\left(g(\boldsymbol{\sigma}) - g_{j,i}(\boldsymbol{\sigma})\right) \\
&\leq \sum_{j=1}^{n}\sum_{i=1}^{m}\left(g(\boldsymbol{\sigma}) - g(\boldsymbol{\sigma}) + \boldsymbol{\sigma}_{j,i}f_j^\star(s_i) + \left|f_j^\star(s_i)\right|\right) \\
&= \sum_{j=1}^{n}\sum_{i=1}^{m}\left(\boldsymbol{\sigma}_{j,i}f_j^\star(s_i) + \left|f_j^\star(s_i)\right|\right) \\
&= g(\boldsymbol{\sigma}) + \sum_{j=1}^{n}\sum_{i=1}^{m}\left|f_j^\star(s_i)\right| \\
&\leq g(\boldsymbol{\sigma}) + n\sup_{f\in\mathcal{F}}\left\{\sum_{i=1}^{m}\left|f(s_i)\right|\right\} \\
&= g(\boldsymbol{\sigma}) + nm\hat{\nu}_{\mathcal{F}}\left(\mathcal{S}\right) \ , \qquad (7.22)
\end{aligned}$$

concluding the proof. □

Theorem 7.5.2. *Let a $n \times m$ matrix $\boldsymbol{\sigma} \in \{-1,1\}^{n\times m}$, and define the function $g(\boldsymbol{\sigma})$ as*

$$g(\boldsymbol{\sigma}) \doteq nm\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) \ .$$

Then $g(\boldsymbol{\sigma})$ is a weakly $(2\hat{z}\left(\mathcal{S}\right), 2nm\hat{\sigma}^2_{\mathcal{F}}\left(\mathcal{S}\right))$-self-bounding function.

Proof. Denote $g_{j,i}(\boldsymbol{\sigma})$ as in the proof of Theorem 7.5.1. To prove that $g(\boldsymbol{\sigma})$ is a weakly (α,β)-self-bounding, we have to prove that, for all $\boldsymbol{\sigma}$, it holds

$$\sum_{j=1}^{n}\sum_{i=1}^{m}\left(g(\boldsymbol{\sigma}) - g_{j,i}(\boldsymbol{\sigma})\right)^2 \leq \alpha g(\boldsymbol{\sigma}) + \beta \ .$$

From the proof of Theorem 7.5.1, we have already proved that

$$g_{j,i}(\boldsymbol{\sigma}) \geq g(\boldsymbol{\sigma}) - \boldsymbol{\sigma}_{j,i}f_j^\star(s_i) - \left|f_j^\star(s_i)\right| \geq g(\boldsymbol{\sigma}) - 2\hat{z}\left(\mathcal{S}\right) \ .$$

Therefore, we observe that

$$\begin{aligned}
&\sum_{j=1}^{n}\sum_{i=1}^{m}\left(g(\boldsymbol{\sigma})-g_{j,i}(\boldsymbol{\sigma})\right)^2 \\
&\leq \sum_{j=1}^{n}\sum_{i=1}^{m}\left(\boldsymbol{\sigma}_{j,i}f_j^\star(s_i)+\left|f_j^\star(s_i)\right|\right)^2 \\
&= \sum_{j=1}^{n}\sum_{i=1}^{m}\left(f_j^\star(s_i)^2+\left|f_j^\star(s_i)\right|^2+2\boldsymbol{\sigma}_{j,i}f_j^\star(s_i)\left|f_j^\star(s_i)\right|\right) \\
&= \sum_{j=1}^{n}\sum_{i=1}^{m}\left(2f_j^\star(s_i)^2+2\boldsymbol{\sigma}_{j,i}f_j^\star(s_i)\left|f_j^\star(s_i)\right|\right) \\
&\leq 2\hat{z}\left(\mathcal{S}\right)\sum_{j=1}^{n}\sum_{i=1}^{m}\boldsymbol{\sigma}_{j,i}f_j^\star(s_i)+2\sum_{j=1}^{n}\sum_{i=1}^{m}f_j^\star(s_i)^2 \\
&= 2\hat{z}\left(\mathcal{S}\right)g(\boldsymbol{\sigma})+2\sum_{j=1}^{n}\sum_{i=1}^{m}f_j^\star(s_i)^2 \\
&\leq 2\hat{z}\left(\mathcal{S}\right)g(\boldsymbol{\sigma})+2n\sup_{f\in\mathcal{F}}\left\{\sum_{i=1}^{m}f(s_i)^2\right\} \\
&= 2\hat{z}\left(\mathcal{S}\right)g(\boldsymbol{\sigma})+2nm\hat{\sigma}_{\mathcal{F}}^2\left(\mathcal{S}\right) \quad ,
\end{aligned}$$

obtaining the statement. □

Theorem 7.5.3. *Let $\boldsymbol{\sigma}\in\{-1,1\}^{n\times m}$ be an $n\times m$ matrix of Rademacher random variables, such that $\boldsymbol{\sigma}_{j,i}\in\{-1,1\}$ independently and with equal probability. Then, for all $0<\varepsilon\leq\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right)$,*

$$\Pr\left(\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right)\geq\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})+\varepsilon\right)\leq\exp\left(-\frac{nm\varepsilon^2}{4\hat{z}\left(\mathcal{S}\right)\left(\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right)+\hat{\nu}_{\mathcal{F}}\left(\mathcal{S}\right)\right)}\right) \quad . \tag{7.5}$$

Proof. Define the set of functions

$$\mathcal{F}'\doteq\left\{f'(x)\doteq f(x)/(2\hat{z}\left(\mathcal{S}\right)):\forall x\in\mathcal{X},f\in\mathcal{F}\right\} \quad ,$$

composed by all functions $f\in\mathcal{F}$ divided by $2\hat{z}\left(\mathcal{S}\right)$; clearly, $|f'(s)|\leq 1/2$, $\forall s\in\mathcal{S}$. We now show that $nm\hat{\mathsf{R}}_m^n(\mathcal{F}',\mathcal{S},\boldsymbol{\sigma})$ (consequently, also $nm\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})$) is a non-negative function:

$$nm\hat{\mathsf{R}}_m^n(\mathcal{F}',\mathcal{S},\boldsymbol{\sigma})\doteq\sum_{j=1}^{n}\sup_{f'\in\mathcal{F}'}\sum_{s_i\in\mathcal{S}}\boldsymbol{\sigma}_{j,i}f'(s_i)\geq\sum_{j=1}^{n}\sum_{s_i\in\mathcal{S}}\boldsymbol{\sigma}_{j,i}f_0(s_i)=0 \quad .$$

From Theorem 7.5.1, we have that $nm\hat{\mathsf{R}}_m^n(\mathcal{F}',\mathcal{S},\boldsymbol{\sigma})$ is a $(1,mn\hat{\nu}_{\mathcal{F}'}\left(\mathcal{S}\right))$-self-bounding function. This implies that it is also a weakly $(1,mn\hat{\nu}_{\mathcal{F}'}\left(\mathcal{S}\right))$-self-bounding function.

Then, note that $\mathbb{E}_{\boldsymbol{\sigma}}\left[nm\hat{\mathsf{R}}_m^n(\mathcal{F}',\mathcal{S},\boldsymbol{\sigma})\right] = nm\hat{\mathsf{R}}\left(\mathcal{F}',\mathcal{S}\right)$. We combine these facts with Theorem 7.3.5, obtaining, for $g(\boldsymbol{\sigma}) = nm\hat{\mathsf{R}}_m^n(\mathcal{F}',\mathcal{S},\boldsymbol{\sigma})$,

$$\Pr\left(nm\hat{\mathsf{R}}\left(\mathcal{F}',\mathcal{S}\right) \geq nm\hat{\mathsf{R}}_m^n(\mathcal{F}',\mathcal{S},\boldsymbol{\sigma}) + t\right) \leq \exp\left(-\frac{t^2}{2\left(nm\hat{\mathsf{R}}\left(\mathcal{F}',\mathcal{S}\right) + nm\hat{\nu}_{\mathcal{F}'}\left(\mathcal{S}\right)\right)}\right) \ .$$

We observe that $\hat{\mathsf{R}}\left(\mathcal{F}',\mathcal{S}\right) = \hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right)/(2\hat{z}\left(\mathcal{S}\right))$, $\hat{\mathsf{R}}_m^n(\mathcal{F}',\mathcal{S},\boldsymbol{\sigma}) = \hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})/(2\hat{z}\left(\mathcal{S}\right))$, and that $\hat{\nu}_{\mathcal{F}'}\left(\mathcal{S}\right) = \hat{\nu}_{\mathcal{F}}\left(\mathcal{S}\right)/(2\hat{z}\left(\mathcal{S}\right))$. We make these substitutions, obtaining

$$\Pr\left(\frac{nm}{2\hat{z}\left(\mathcal{S}\right)}\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) \geq \frac{nm}{2\hat{z}\left(\mathcal{S}\right)}\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + t\right) \leq \exp\left(-\frac{\hat{z}\left(\mathcal{S}\right)t^2}{\left(nm\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) + nm\hat{\nu}_{\mathcal{F}}\left(\mathcal{S}\right)\right)}\right) \ .$$

We further substitute t by $nm\varepsilon/(2\hat{z}\left(\mathcal{S}\right))$, obtaining the statement. □

Theorem 7.5.4. *Let $\boldsymbol{\sigma} \in \{-1,1\}^{n\times m}$ be an $n \times m$ matrix of Rademacher random variables, such that $\boldsymbol{\sigma}_{j,i} \in \{-1,1\}$ independently and with equal probability. Then, for all $0 < \varepsilon \leq \hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right)$,*

$$\Pr\left(\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) \geq \hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \varepsilon\right) \leq \exp\left(-\frac{nm\varepsilon^2}{4\left(\hat{z}\left(\mathcal{S}\right)\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) + \hat{\sigma}_{\mathcal{F}}^2\left(\mathcal{S}\right)\right)}\right) \ . \quad (7.6)$$

Proof. Let $\mathcal{F}'$ be the same set of functions defined in the proof of Theorem 7.5.3. If we denote $g(\boldsymbol{\sigma}) \doteq nm\hat{\mathsf{R}}_m^n(\mathcal{F}',\mathcal{S},\boldsymbol{\sigma})$, then, from Theorem 7.5.2, $g(\boldsymbol{\sigma})$ is a weakly $(1, 2mn\hat{\sigma}_{\mathcal{F}'}^2\left(\mathcal{S}\right))$-self-bounding function. As before, $\mathbb{E}_{\boldsymbol{\sigma}}\left[nm\hat{\mathsf{R}}_m^n(\mathcal{F}',\mathcal{S},\boldsymbol{\sigma})\right] = nm\hat{\mathsf{R}}\left(\mathcal{F}',\mathcal{S}\right)$. We apply Theorem 7.3.5 on $g(\boldsymbol{\sigma}) = nm\hat{\mathsf{R}}_m^n(\mathcal{F}',\mathcal{S},\boldsymbol{\sigma})$, obtaining

$$\Pr\left(nm\hat{\mathsf{R}}\left(\mathcal{F}',\mathcal{S}\right) \geq nm\hat{\mathsf{R}}_m^n(\mathcal{F}',\mathcal{S},\boldsymbol{\sigma}) + t\right) \leq \exp\left(-\frac{t^2}{2\left(nm\hat{\mathsf{R}}\left(\mathcal{F}',\mathcal{S}\right) + 2nm\hat{\sigma}_{\mathcal{F}'}^2\left(\mathcal{S}\right)\right)}\right) \ .$$

We observe that $\hat{\mathsf{R}}\left(\mathcal{F}',\mathcal{S}\right) = \hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right)/(2\hat{z}\left(\mathcal{S}\right))$, $\hat{\mathsf{R}}_m^n(\mathcal{F}',\mathcal{S},\boldsymbol{\sigma}) = \hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})/(2\hat{z}\left(\mathcal{S}\right))$, and that $\hat{\sigma}_{\mathcal{F}'}^2\left(\mathcal{S}\right) = \hat{\sigma}_{\mathcal{F}}^2\left(\mathcal{S}\right)/(4\hat{z}\left(\mathcal{S}\right)^2)$. This implies that

$$\Pr\left(\frac{nm}{2\hat{z}\left(\mathcal{S}\right)}\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) \geq \frac{nm}{2\hat{z}\left(\mathcal{S}\right)}\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + t\right) \leq \exp\left(-\frac{t^2}{\left(\frac{nm}{\hat{z}(\mathcal{S})}\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) + \frac{nm}{\hat{z}(\mathcal{S})^2}\hat{\sigma}_{\mathcal{F}}^2\left(\mathcal{S}\right)\right)}\right) \ .$$

Replacing t by $\varepsilon nm/(2\hat{z}\left(\mathcal{S}\right))$ concludes the proof. □

Theorem 7.5.5. *With probability $\geq 1-\delta$ it holds*

$$\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) \leq \hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \frac{2\hat{z}\left(\mathcal{S}\right)\ln\left(\frac{1}{\delta}\right)}{nm} + \sqrt{\left(\frac{2\hat{z}\left(\mathcal{S}\right)\ln\left(\frac{1}{\delta}\right)}{nm}\right)^2 + \frac{4\hat{z}\left(\mathcal{S}\right)\left(\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \hat{\nu}_{\mathcal{F}}\left(\mathcal{S}\right)\right)\ln\left(\frac{1}{\delta}\right)}{nm}} \ .$$

Also, with probability $\geq 1-\delta$, it holds

$$\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) \leq \hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \frac{2\hat{z}\left(\mathcal{S}\right)\ln\left(\frac{1}{\delta}\right)}{nm} + \sqrt{\left(\frac{2\hat{z}\left(\mathcal{S}\right)\ln\left(\frac{1}{\delta}\right)}{nm}\right)^2 + \frac{4\left(\hat{z}\left(\mathcal{S}\right)\hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \hat{\sigma}_{\mathcal{F}}^2\left(\mathcal{S}\right)\right)\ln\left(\frac{1}{\delta}\right)}{nm}} \ .$$

Proof. We prove the first inequality, as proving the second is analogous. From Theorem 7.5.3, we have that, with probability $\geq 1-\delta$,

$$\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) \leq \hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \sqrt{\frac{4\hat{z}\left(\mathcal{S}\right)\left(\hat{\nu}_{\mathcal{F}}\left(\mathcal{S}\right) + \hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right)\right)\ln\left(\frac{1}{\delta}\right)}{nm}} \ .$$

An upper bound to $\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right)$ can be obtained by finding the fixed point of the function $r(x)$

$$r(x) \doteq \hat{\mathsf{R}}_m^n(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \sqrt{\frac{4\hat{z}\left(\mathcal{S}\right)\left(\hat{\nu}_{\mathcal{F}}\left(\mathcal{S}\right) + x\right)\ln\left(\frac{1}{\delta}\right)}{nm}} \ .$$

In fact, it is trivial to prove the following.

Lemma 7.8.1. *Let $u, v, y \geq 0$. The fixed point of*

$$r(x) = u + \sqrt{v + yx}$$

is at

$$x = u + \frac{y}{2} + \sqrt{\frac{y^2}{4} + uy + v} \ .$$

Thus, we apply Lemma 7.8.1 to obtain, after simple calculations, the statement. □

Theorem 7.5.7. *With probability $\geq 1-\delta$, it holds*

$$\mathsf{R}(\mathcal{F},m) \leq \hat{\mathsf{R}}^1_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \sqrt{\frac{2(2z\mathsf{R}(\mathcal{F},m)+\sigma^2_{\mathcal{F}})\ln\left(\frac{1}{\delta}\right)}{m}} + \frac{z\ln\left(\frac{1}{\delta}\right)}{8m} \tag{7.7}$$

$$\leq \hat{\mathsf{R}}^1_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) + \sqrt{\frac{9}{8}\left(\frac{2z\ln\left(\frac{1}{\delta}\right)}{m}\right)^2 + \frac{2(2z\hat{\mathsf{R}}^1_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})+\sigma^2_{\mathcal{F}})\ln\left(\frac{1}{\delta}\right)}{m}} + \frac{17z\ln\left(\frac{1}{\delta}\right)}{8m} \ . \tag{7.8}$$

Proof. Define the set of functions $\mathcal{G}$ as

$$\mathcal{G} \doteq \{g : g(x,\sigma) \doteq \sigma f(x), f \in \mathcal{F}, x \in \mathcal{X}, \sigma \in \{-1,1\}\} \ ,$$

where σ is a Rademacher random variable. Therefore, we observe that, from independence of the random variables σ and $f(x)$,

$$\mathbb{E}[g] = \mathbb{E}[f]\mathbb{E}[\sigma] = 0 \ , \ \ \|g\|_\infty = \sup\{|\sigma f(x)| : \sigma \in \{-1,1\}, x \in \mathcal{X}, f \in \mathcal{F}\} \leq z \ ,$$

$$\begin{aligned}\sup_{g\in\mathcal{G}} Var(g) &= \sup_{f\in\mathcal{F}}\left\{\mathbb{E}\left[\sigma^2\right]\mathbb{E}\left[f^2\right] - \left(\mathbb{E}\left[\sigma\right]\mathbb{E}\left[f\right]\right)^2\right\} \\ &= \sup_{f\in\mathcal{F}}\left\{\mathbb{E}\left[\sigma^2\right]\mathbb{E}\left[f^2\right]\right\} = \sup_{f\in\mathcal{F}}\left\{\mathbb{E}\left[f^2\right]\right\} = \sigma^2_{\mathcal{F}} \ .\end{aligned}$$

We now need the following left tail bound of Bousquet's inequality.

Corollary 7.8.2 (Corollary 12.2, Boucheron et al. (2013))**.** *Consider the setup of Theorem 7.6.1. Then, for all $t \geq 0$, it holds*

$$\Pr\left(Z \leq \mathbb{E}[Z] - \sqrt{2vt} - \frac{dt}{8}\right) \leq \exp(-t) \ .$$

Thus, we apply Corollary 7.8.2 to $\mathcal{G}$ to obtain (7.7). The bound of (7.8) follows from Lemma 7.8.1. □

Theorem 7.5.8. *It holds $\sigma^2_{\mathcal{F}} \leq \mathbb{E}\left[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right]$, and, for $\varepsilon \leq \mathbb{E}\left[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right]$,*

$$\Pr\left(\mathbb{E}\left[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right] \geq \hat{\sigma}^2_{\mathcal{F}}(\mathcal{S}) + \varepsilon\right) \leq \exp\left(-\frac{m\mathbb{E}\left[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right]}{z^2} h\left(-\frac{\varepsilon}{\mathbb{E}\left[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right]}\right)\right) \tag{7.9}$$

$$\leq \exp\left(-\frac{m\varepsilon^2}{2z^2\mathbb{E}\left[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})\right]}\right) \ . \tag{7.10}$$

Furthermore, with probability $\geq 1-\delta$*, it holds*

$$\sigma_{\mathcal{F}}^2 \leq \hat{\sigma}_{\mathcal{F}}^2(\mathcal{S}) + \frac{z^2 \ln\left(\frac{1}{\delta}\right)}{m} + \sqrt{\left(\frac{z^2 \ln\left(\frac{1}{\delta}\right)}{m}\right)^2 + \frac{2z^2\hat{\sigma}_{\mathcal{F}}^2(\mathcal{S}) \ln\left(\frac{1}{\delta}\right)}{m}} \ . \tag{7.11}$$

Proof. We first prove that

$$\sigma_{\mathcal{F}}^2 \leq \mathbb{E}\left[\hat{\sigma}_{\mathcal{F}}^2(\mathcal{S})\right] \ .$$

Through Jensen's inequality, we have

$$\begin{aligned}\sigma_{\mathcal{F}}^2 = \sup_{f\in\mathcal{F}}\left\{\mathbb{E}\left[f^2\right]\right\} &= \sup_{f\in\mathcal{F}}\left\{\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}\left(f(s_i)\right)^2\right]\right\} \\ &\leq \mathbb{E}\left[\sup_{f\in\mathcal{F}}\left\{\frac{1}{m}\sum_{i=1}^{m}\left(f(s_i)\right)^2\right\}\right] = \mathbb{E}\left[\hat{\sigma}_{\mathcal{F}}^2(\mathcal{S})\right] \ .\end{aligned}$$

We now show that $\hat{\sigma}_{\mathcal{F}}^2(\mathcal{S})$ is a $(1,0)$-self-bounding function. Let the function $g(\mathcal{S}) = m\hat{\sigma}_{\mathcal{F}}^2(\mathcal{S})$, and, for $j \in [1,m]$, let the function $g_j(\mathcal{S})$ be

$$g_j(\mathcal{S}) = \sup_{f\in\mathcal{F}}\left\{\sum_{\substack{i=1\\ i\neq j}}^{m}\left(f(s_i)\right)^2\right\} \ .$$

First, it holds $g(\mathcal{S}) \geq 0$, and $g_j(\mathcal{S}) \leq g(\mathcal{S})$, for all $\mathcal{S}$ and all j, as $\left(f(s)\right)^2 \geq 0, \forall s$. We now prove that $g(\mathcal{S}) - g_j(\mathcal{S}) \leq z^2$. Let $f^\star$ be one of the functions of $\mathcal{F}$ attaining the supremum for $g(\mathcal{S})$; then,

$$\begin{aligned}g_j(\mathcal{S}) = \sup_{f\in\mathcal{F}}\left\{\sum_{\substack{i=1\\ i\neq j}}^{m}\left(f(s_i)\right)^2\right\} \geq \sum_{\substack{i=1\\ i\neq j}}^{m}\left(f^\star(s_i)\right)^2 &= \sum_{i=1}^{m}\left(f^\star(s_i)\right)^2 - \left(f^\star(s_j)\right)^2 \\ &= g(\mathcal{S}) - \left(f^\star(s_j)\right)^2 \geq g(\mathcal{S}) - z^2 \ .\end{aligned}$$

Consequently, we have

$$\sum_{j=1}^{m}\left(g(\mathcal{S}) - g_j(\mathcal{S})\right) \leq \sum_{j=1}^{m}\left(\left(f^\star(s_j)\right)^2\right) = g(\mathcal{S}) \ ,$$

that concludes the proof that $\hat{\sigma}_{\mathcal{F}}^2(\mathcal{S})$ is a $(1,0)$-self-bounding function (we have ignored the scaling of $1/z^2$, but analogous steps of the proof of Theorem 7.5.3 are easy to follow). We now apply Theorem 7.3.6 to obtain a probabilistic bounds to the

expectation $\mathbb{E}[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})]$ of $\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})$; we have, for $\varepsilon \leq \mathbb{E}[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})]$,

$$\Pr\left(\mathbb{E}[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})] \geq \hat{\sigma}^2_{\mathcal{F}}(\mathcal{S}) + \varepsilon\right) \leq \exp\left(-\frac{\mathbb{E}[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})]}{z^2} h\left(-\frac{\varepsilon}{\mathbb{E}[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})]}\right)\right) \ ,$$

obtaining (7.9). The inequality (7.10) is a consequence of the fact that $h(-x) \geq x^2/2, \forall x \in [0,1]$, as pointed out by Boucheron et al. (2000). Then, (7.11) follows from bounding the rightmost term of (7.10) below δ, by applying Lemma 7.8.1, and from $\sigma^2_{\mathcal{F}} \leq \mathbb{E}[\hat{\sigma}^2_{\mathcal{F}}(\mathcal{S})]$. □

Theorem 7.7.1. *Assume $c \leq 1$. Let $g(\mathcal{S})$ be*

$$g(\mathcal{S}) \doteq m\mathsf{D}^+(\mathcal{F}, \mathcal{S}) = \sup_{f \in \mathcal{F}} \left\{ \sum_{j=1}^{m} f(s_j) - m\mathbb{E}[f] \right\} \ .$$

Then, $g(\mathcal{S})$ is a $(1, m\eta_{\mathcal{F}})$-self-bounding function.

Proof. Let $g_i(\mathcal{S})$ be

$$g_i(\mathcal{S}) = \inf_{s'_i} \left\{ \sup_{f \in \mathcal{F}} \left\{ \sum_{\substack{j=1 \\ j \neq i}}^{m} f(s_j) + f(s'_i) - m\mathbb{E}[f] \right\} \right\} \ .$$

Notice that, as done before, s_i is ignored in the definition of $g_i(\mathcal{S})$. Let $f^\star$ be one of the functions in $\mathcal{F}$ that attains the supremum for $g(\mathcal{S})$. We then have

$$\begin{aligned}
g_i(\mathcal{S}) &= \inf_{s'_i} \left\{ \sup_{f \in \mathcal{F}} \left\{ \sum_{\substack{j=1 \\ j \neq i}}^{m} f(s_j) + f(s'_i) - m\mathbb{E}[f] \right\} \right\} \\
&\geq \inf_{s'_i} \left\{ \sum_{\substack{j=1 \\ j \neq i}}^{m} f^\star(s_j) + f^\star(s'_i) - m\mathbb{E}[f^\star] \right\} \\
&= \sum_{\substack{j=1 \\ j \neq i}}^{m} f^\star(s_j) - m\mathbb{E}[f^\star] + \inf_{s'_i} \{f^\star(s'_i)\} \\
&= \sum_{\substack{j=1 \\ j \neq i}}^{m} f^\star(s_j) - m\mathbb{E}[f^\star] + a \\
&= \sum_{\substack{j=1 \\ j \neq i}}^{m} f^\star(s_j) + f^\star(s_i) - f^\star(s_i) - m\mathbb{E}[f^\star] + a \\
&= g(\mathcal{S}) - f^\star(s_i) + a \ .
\end{aligned}$$

We then observe that $g_i(\mathcal{S}) \geq g(\mathcal{S}) - b + a = g(\mathcal{S}) - c$; assuming that $c \leq 1$, we have $g_i(\mathcal{S}) \geq g(\mathcal{S}) - 1$. We then continue with

$$\begin{aligned}
&\sum_{j=1}^{m} (g(\mathcal{S}) - g_i(\mathcal{S})) \\
&\leq \sum_{j=1}^{m} (f^\star(s_i) - a) \\
&= \sum_{j=1}^{m} f^\star(s_i) - am \\
&= \sum_{j=1}^{m} f^\star(s_i) - m\mathbb{E}[f^\star] + m\mathbb{E}[f^\star] - am \\
&= g(\mathcal{S}) + m\mathbb{E}[f^\star] - ma \\
&\leq g(\mathcal{S}) + m\eta_{\mathcal{F}} \ ,
\end{aligned}$$

obtaining the statement. □

Theorem 7.7.2. *Assume $c \leq 1$. Let $g(\mathcal{S})$ be*

$$g(\mathcal{S}) \doteq m\mathsf{D}^-(\mathcal{F}, \mathcal{S}) = \sup_{f \in \mathcal{F}} \left\{ m\mathbb{E}[f] - \sum_{j=1}^{m} f(s_j) \right\} \ .$$

Then, $g(\mathcal{S})$ is a $(1, m\gamma_{\mathcal{F}})$-self-bounding function.

Proof. Define the set of functions $\mathcal{F}' \doteq \{f'(x) \doteq -f(x), f \in \mathcal{F}, x \in \mathcal{X}\}$. We have that $f' \in [-b, -a]$, that $\mathbb{E}[f'] = -\mathbb{E}[f]$, and that $\sum_{j=1}^{m} f'(s_j) = -\sum_{j=1}^{m} f(s_j)$. Therefore,

$$g(\mathcal{S}) = \sup_{f \in \mathcal{F}} \left\{ m\mathbb{E}[f] - \sum_{j=1}^{m} f(s_j) \right\} = \sup_{f' \in \mathcal{F}'} \left\{ \sum_{j=1}^{m} f'(s_j) - m\mathbb{E}[f'] \right\} \ .$$

Then, we may observe that

$$\gamma_{\mathcal{F}} = b - \inf_{f \in \mathcal{F}} \mathbb{E}[f] = b + \sup_{f \in \mathcal{F}} \mathbb{E}[f'] = \sup_{f \in \mathcal{F}} \mathbb{E}[f'] - \min_{x} f'(x) \ .$$

Thus, we apply Theorem 7.7.1 to $g(\mathcal{S})$ and $\mathcal{F}'$ to show that it is $(1, m\gamma_{\mathcal{F}})$-self bounding, obtaining the statement. □

Theorem 7.7.3. *Let Z be*

$$Z \doteq \mathsf{D}^+(\mathcal{F}, \mathcal{S}) = \sup_{f \in \mathcal{F}} \left\{ \frac{1}{m} \sum_{j=1}^{m} f(s_j) - \mathbb{E}[f] \right\} \ .$$

Then, it holds

$$\Pr\left(Z \geq \mathbb{E}\left[Z\right] + \varepsilon\right) \leq \exp\left(-\frac{m\varepsilon^2}{2c\left(\mathbb{E}\left[Z\right] + \eta_{\mathcal{F}} + \varepsilon/3\right)}\right). \tag{7.14}$$

Consequently, with probability $\geq 1-\delta$,

$$Z \leq \mathbb{E}\left[Z\right] + \sqrt{\left(\frac{c\ln\left(\frac{1}{\delta}\right)}{3m}\right)^2 + \frac{2c\ln\left(\frac{1}{\delta}\right)\left(\mathbb{E}\left[Z\right] + \eta_{\mathcal{F}}\right)}{m}} + \frac{c\ln\left(\frac{1}{\delta}\right)}{3m} \ . \tag{7.15}$$

Proof. We first observe that $g(\mathcal{S})$ is a non-negative function, for all $\mathcal{S}$, since $f_0 \in \mathcal{F}$. Then, $g(\mathcal{S}) \doteq mZ$ is $(1, m\eta_{\mathcal{F}})$-self-bounding from Theorem 7.7.1; therefore, we apply Theorem 7.3.5 to obtain (7.14). The second statement follows from imposing the r.h.s. of (7.14) to be $\leq \delta$. □

Theorem 7.7.4. *Let* Z *be*

$$Z \doteq \mathsf{D}^{-}(\mathcal{F}, \mathcal{S}) = \sup_{f\in\mathcal{F}}\left\{\mathbb{E}\left[f\right] - \frac{1}{m}\sum_{j=1}^{m} f(s_j)\right\} \ .$$

Then, it holds

$$\Pr\left(Z \geq \mathbb{E}\left[Z\right] + \varepsilon\right) \leq \exp\left(-\frac{m\varepsilon^2}{2c\left(\mathbb{E}\left[Z\right] + \gamma_{\mathcal{F}} + \varepsilon/3\right)}\right). \tag{7.16}$$

Consequently, with probability $\geq 1-\delta$,

$$Z \leq \mathbb{E}\left[Z\right] + \sqrt{\left(\frac{c\ln\left(\frac{1}{\delta}\right)}{3m}\right)^2 + \frac{2c\ln\left(\frac{1}{\delta}\right)\left(\mathbb{E}\left[Z\right] + \gamma_{\mathcal{F}}\right)}{m}} + \frac{c\ln\left(\frac{1}{\delta}\right)}{3m} \ . \tag{7.17}$$

Proof. We follow analogous steps taken in the proof of Theorem 7.7.3. First, $g(\mathcal{S}) \doteq mZ$ is $(1, m\gamma_{\mathcal{F}})$-self-bounding from Theorem 7.7.2; (7.16) follows from Theorem 7.3.5. The second statement is again obtained from bounding the r.h.s. of (7.16) below δ. □

Theorem 7.7.5. *It holds* $\eta_{\mathcal{F}} \leq \mathbb{E}\left[\hat{\eta}_{\mathcal{F}}\left(\mathcal{S}\right)\right]$, *and, for* $\varepsilon \leq \mathbb{E}\left[\hat{\eta}_{\mathcal{F}}\left(\mathcal{S}\right)\right]$,

$$\begin{aligned}\Pr\left(\mathbb{E}\left[\hat{\eta}_{\mathcal{F}}\left(\mathcal{S}\right)\right] \geq \hat{\eta}_{\mathcal{F}}\left(\mathcal{S}\right) + \varepsilon\right) &\leq \exp\left(-\frac{m\mathbb{E}\left[\hat{\eta}_{\mathcal{F}}\left(\mathcal{S}\right)\right]}{c} h\left(-\frac{\varepsilon}{\mathbb{E}\left[\hat{\eta}_{\mathcal{F}}\left(\mathcal{S}\right)\right]}\right)\right) \\ &\leq \exp\left(-\frac{m\varepsilon^2}{2c\mathbb{E}\left[\hat{\eta}_{\mathcal{F}}\left(\mathcal{S}\right)\right]}\right) \ .\end{aligned} \tag{7.19}$$

Furthermore, with probability $\geq 1-\delta$*, it holds*

$$\eta_{\mathcal{F}} \leq \hat{\eta}_{\mathcal{F}}(\mathcal{S}) + \frac{c\ln\left(\frac{1}{\delta}\right)}{m} + \sqrt{\left(\frac{c\ln\left(\frac{1}{\delta}\right)}{m}\right)^2 + \frac{2c\hat{\eta}_{\mathcal{F}}(\mathcal{S})\ln\left(\frac{1}{\delta}\right)}{m}} \ .$$

Proof. We follow similar steps taken in the proof of Theorem 7.5.8. We first prove that

$$\eta_{\mathcal{F}} \leq \mathbb{E}\left[\hat{\eta}_{\mathcal{F}}(\mathcal{S})\right]$$

by observing, through Jensen's inequality, that

$$\begin{aligned}\eta_{\mathcal{F}} = \sup_{f\in\mathcal{F}}\left\{\mathbb{E}\left[f\right]\right\} - a &= \sup_{f\in\mathcal{F}}\left\{\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} f(s_i)\right]\right\} - a \\ &\leq \mathbb{E}\left[\sup_{f\in\mathcal{F}}\left\{\frac{1}{m}\sum_{i=1}^{m} f(s_i)\right\}\right] - a = \mathbb{E}\left[\hat{\eta}_{\mathcal{F}}(\mathcal{S})\right] \ .\end{aligned}$$

We now show that $\hat{\eta}_{\mathcal{F}}(\mathcal{S})$ is a self-bounding function. Let the function $g(\mathcal{S}) = m\hat{\eta}_{\mathcal{F}}(\mathcal{S})$, and, for $j \in [1,m]$, let the function $g_j(\mathcal{S})$ be

$$g_j(\mathcal{S}) = \inf_{s'_j}\left\{\sup_{f\in\mathcal{F}}\left\{\sum_{\substack{i=1\\ i\neq j}}^{m} f(s_i) + f\left(s'_j\right)\right\}\right\} - a \ .$$

First, it holds $g(\mathcal{S}) \geq 0$, as $f(s) \geq a, \forall s$, and $g_j(\mathcal{S}) \leq g(\mathcal{S})$ by definition of $g_j(\mathcal{S})$. We now prove that $g(\mathcal{S}) - g_j(\mathcal{S}) \leq c$. Let $f^\star$ be one of the functions of $\mathcal{F}$ attaining the supremum for $g(\mathcal{S})$; then,

$$\begin{aligned}g_j(\mathcal{S}) = \inf_{s'_j}\left\{\sup_{f\in\mathcal{F}}\left\{\sum_{\substack{i=1\\ i\neq j}}^{m} f(s_i) + f\left(s'_j\right)\right\}\right\} - a &\geq \sum_{\substack{i=1\\ i\neq j}}^{m} f^\star(s_i) + \inf_{s'_j}\left\{f^\star\left(s'_j\right)\right\} - a \\ &= \sum_{i=1}^{m} f^\star(s_i) - f^\star(s_j) \\ &= g(\mathcal{S}) - f^\star(s_j) + a \geq g(\mathcal{S}) - c \ .\end{aligned}$$

Consequently, we have

$$\sum_{j=1}^{m}\left(g(\mathcal{S}) - g_j(\mathcal{S})\right) \leq \sum_{j=1}^{m}\left(f^\star(s_j) - a\right) = g(\mathcal{S}) \ ,$$

that concludes the proof that $\hat{\eta}_{\mathcal{F}}(\mathcal{S})$ is a $(1,0)$-self-bounding function. We now apply

Theorem 7.3.6 to a family of functions that is scaled by $1/c$ (i.e., as we did in the proof of Theorem 7.5.3) to obtain a probabilistic bounds to the expectation $\mathbb{E}[\hat{\eta}_{\mathcal{F}}(\mathcal{S})]$ of $\hat{\eta}_{\mathcal{F}}(\mathcal{S})$; we have

$$\Pr\left(\mathbb{E}[\hat{\eta}_{\mathcal{F}}(\mathcal{S})] \geq \hat{\eta}_{\mathcal{F}}(\mathcal{S}) + \varepsilon\right) \leq \exp\left(-\frac{\mathbb{E}[\hat{\eta}_{\mathcal{F}}(\mathcal{S})]}{c} h\left(-\frac{\varepsilon}{\mathbb{E}[\hat{\eta}_{\mathcal{F}}(\mathcal{S})]}\right)\right) \quad ,$$

proving (7.19). The rest follows analogously as in the proof of Theorem 7.5.8. □

Theorem 7.7.6. *It holds $\gamma_{\mathcal{F}} \leq \mathbb{E}\left[\hat{\gamma}_{\mathcal{F}}(\mathcal{S})\right]$, and, for $\varepsilon \leq \mathbb{E}\left[\hat{\gamma}_{\mathcal{F}}(\mathcal{S})\right]$,*

$$\begin{aligned}\Pr\left(\mathbb{E}\left[\hat{\gamma}_{\mathcal{F}}(\mathcal{S})\right] \geq \hat{\gamma}_{\mathcal{F}}(\mathcal{S}) + \varepsilon\right) &\leq \exp\left(-\frac{m\mathbb{E}\left[\hat{\gamma}_{\mathcal{F}}(\mathcal{S})\right]}{c} h\left(-\frac{\varepsilon}{\mathbb{E}\left[\hat{\gamma}_{\mathcal{F}}(\mathcal{S})\right]}\right)\right) \\ &\leq \exp\left(-\frac{m\varepsilon^2}{2c\mathbb{E}\left[\hat{\gamma}_{\mathcal{F}}(\mathcal{S})\right]}\right) \quad .\end{aligned}$$

Furthermore, with probability $\geq 1-\delta$, it holds

$$\gamma_{\mathcal{F}} \leq \hat{\gamma}_{\mathcal{F}}(\mathcal{S}) + \frac{c\ln\left(\frac{1}{\delta}\right)}{m} + \sqrt{\left(\frac{c\ln\left(\frac{1}{\delta}\right)}{m}\right)^2 + \frac{2c\hat{\gamma}_{\mathcal{F}}(\mathcal{S})\ln\left(\frac{1}{\delta}\right)}{m}} \quad .$$

Proof. First, we define the set of functions $\mathcal{F}'$ as

$$\mathcal{F}' = \{f' : f'(x) = -f(x), f \in \mathcal{F}, x \in \mathcal{X}\} \quad ,$$

and we define $a' = \inf_x f'(x) = -b$, $b' = \sup_x f'(x) = -a$. We have $\eta_{\mathcal{F}'} = \sup_{f'} \mathbb{E}[f'] - a' = \gamma_{\mathcal{F}}$, and $\hat{\eta}_{\mathcal{F}'}(\mathcal{S}) = \hat{\gamma}_{\mathcal{F}}(\mathcal{S})$. Thus, we apply Theorem 7.7.5 and Lemma 7.8.1 to $\mathcal{F}'$, obtaining, after appropriate substitutions, all the statements of the Theorem for $\mathcal{F}$. □

Chapter 8

Conclusions

In this Chapter we summarize the contributions of this Thesis, and discuss future research directions.

In Chapter 3, we presented TopKWY, a novel algorithm to mine the *Top-k Significant Patterns* with rigorous control of false discoveries. Focusing on the most significant patterns allows to bound the size of the output below k, resulting in significant computational advantages, but without sacrificing the guaranteed statistical significance of the set of reported results. The key to the efficiency of TopKWY is the combination of a novel exploration strategy of the search space of patterns, that we prove will never explore non-interesting candidates, novel bounds to skip the processing of many explored candidates, and the powerful Westfall-Young (WY) permutation testing framework. We also show that simple modifications to the WY procedure lead to variants of TopKWY to control more flexible error rates, such as the Generalized Family-Wise Error Rate, and the False Discovery Proportion; such rates provide, when needed, strong improvements in terms of *power* of the procedure, without losing control of the overall size of the output.

In Chapter 4 we introduced SPuManTE, an efficient algorithm for Significant Pattern Mining based on Unconditional Testing. Unconditional tests are better suited for Knowledge Discovery settings, as they assume less stringent constraints to the generative process of the data. The use of unconditional tests in practice was limited by computational reasons; in SPuManTE we tackle this challenge with a novel algorithm to efficiently perform the test, and with the combination of recent results on uniformly valid probabilistic confidence intervals to the frequencies of the patterns, leveraging key concepts from Statistical Learning Theory.

In Chapter 5 we described a new sampling-based algorithm, called `SAKEIMA`, to compute approximations of the collection of frequent strings of length k, called k-mers, from massive high-throughput sequencing datasets. The analysis of k-mers is one of the first step in many computational biology pipelines, often very demanding because of the size of the datasets and of the exponential number of distinct k-mers that may appear in them. We have shown that `SAKEIMA`, thanks to its advanced sampling scheme, computes an high-quality rigorous approximation of the set of frequent k-mers by analyzing only a *fraction* of the data. `SAKEIMA` can be applied to speedup analyses based on the abundace of k-mers, such as the computation of approximated distances between sequencing datasets in metagenomics.

In Chapter 6 we presented MCRapper, a novel strategy for the efficient computation of *Monte Carlo Empirical Rademacher Averages (n-MCERA)* for pattern languages with a poset structure. The n-MCERA is a key quantity to derive sharp, data-dependent, uniformly valid confidence bounds on the expectations of sets of functions from random samples. Since many pattern languages satisfy a poset structure, MCRapper has direct and wide applications in Statistically-sound, Significant and Approximate Pattern Mining problems and, given the power and the generality of the n-MCERA, possibly in many other settings. We discussed and tested a specific application of MCRapper in discovering True Frequent Patterns with a novel strat-

egy, called TFP-R, based on variance-aware concentration inequalities and a novel strategy to restrict the computation of the n-MCERA to properly selected subsets of set of functions. We showed that MCRapper and TFP-R significantly outperform other state-of-the-art methods in their respective tasks.

In Chapter 7 we proved that the n-MCERA satisfies certain *self-bounding* properties, important concepts in the theory of concentration of measure inequalities. These properties imply novel variance-dependent concentration bounds of the Monte Carlo estimated value of the n-MCERA w.r.t. its expectation, the Empirical Rademacher Complexity. Our novel bounds represent a strong improvement in the trade-off between n, the number of Monte Carlo trials to perform for the computation of the n-MCERA, and the confidence bounds for its convergence when characteristic quantities of the functions, such as the empirical wimpy-variance, are small. Such bounds are particularly useful in the framework of Localized Rademacher Complexities, where the family of functions under consideration are often restricted to subsets with small variance. As some of these bounds depend on unknown properties of the family of functions under consideration, such as the wimpy variance, we leverage again the powerful framework of self-bounding functions to derive novel concentration inequalities that allow to sharply estimate them from their empirical counterparts. This is the case, for example, of the wimpy variance, that we show can be tightly estimated using the empirical wimpy variance. Then, we prove self-bounding properties for the Supremum Deviations between empirical averages of functions from a family and their expectations, analogues to the ones satisfied by the n-MCERA. Consequently, we proved novel variance-dependent concentration bounds to the Supremum Deviations, that may be of independent interest.

There are many possible extensions of the contributions of this work and new directions for future research. We first discuss possible future investigations in the context of Significant Pattern Mining. The notion of Top-k Significant Patterns we give in Chapter 3 and our algorithm TopKWY could be relevant to other mining problems while controlling false discoveries, such as Statistical Emerging Pattern Mining (Komiyama et al., 2017), Subgroup Discovery (Belfodil et al., 2019, 2018), and expressive languages combining co-occurence and mutual exclusivity (Fischer and Vreeken, 2020). Extensions considering a numeric target, numeric features (Sugiyama and Borgwardt, 2019), correcting confounding effects with covariates (Terada et al., 2016; Papaxanthos et al., 2016; Llinares-López et al., 2017, 2019), and other statistical tests, such as the ones employed in survival analysis (Relator et al., 2018) are also important and interesting directions.

Then, other definitions of Top-k Significant Patterns may be investigated; for example, maximizing *diversity* among the set of Top-k Significant Patterns may be important to reduce potential redundancy in the results. Sophisticated techniques proposed to mine diverse and non-redundant sets of interesting patterns (Van Leeuwen and Knobbe, 2012; Vreeken et al., 2011; Knobbe and Ho, 2006; Kalofolias et al., 2017) and *subjectively interesting* patterns (van Leeuwen et al., 2016) are relevant to address

this issue; efficiently combining such techniques with methods to provide guarantees on the statistical significance of the output is a challenging problem.

While in TopKWY, and its variants, we focused on bounding the $FWER$, the g-$FWER$, and the FDP, a different approach would be to bound the False Discovery Rate (FDR) (Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001), that is the expected ratio of false discoveries among all reported patterns. Furthermore, our solution to control the FDP relies on a modification to the Westfall-Young (WY) permutation testing procedure *and* an upper bound k to the maximum number of results; controlling efficiently the FDP with *unbounded* k is a challenging problem and an interesting direction for future investigations.

In Chapter 4 we argued that *unconditional* assumptions are more natural in Knowledge Discovery settings, since traditionally employed *conditional* assumptions impose very restrictive constraints on the data generative process, and we have shown their difference in practice in the problem of testing hypothesis in Significant Pattern Mining. In addition to this problem, we believe there may be other settings worth investigating in which conditional assumptions are made, sacrificing statistical aspects for computational reasons. Furthermore, combining unconditional tests with different procedures for Multiple Hypothesis Testing, such as WY permutation testing, is another interesting future direction.

Computing approximations of frequent k-mers with sampling has potentially many interesting directions for future research. While, in Chapter 5, we presented results for k-mers from datasets of short reads, `SAKEIMA` may be adapted for the analysis of k-mers with spaced seeds (Břinda et al., 2015), to process large datasets of long reads, and whole genome sequences. Another interesting direction is to modify the sampling strategy of `SAKEIMA`, in order to consider samples of *entire reads* (Santoro et al., 2021) instead of individial occurences of k-mers; this would result in an even more transparent approach to embed into existing k-mer counting tools. The challenging analysis of this modified strategy requires to handle dependencies between k-mers appearing in the same sequence; more sophisticated concepts of complexities, such as the *pseudodimenion* (Pollard, 1984), may be relevant for this problem.

In Chapter 6 we have shown that the Monte-Carlo Empirical Rademacher Average (n-MCERA) has the flexibility and the power needed to compute sharp bounds to the rate of uniform convergence of empirical averages of families of functions for Pattern Mining applications. In general, the n-MCERA has received scant attention since its proposal, probably since computing it efficiently is challenging. We believe that the n-MCERA and the ideas we developed to compute it efficiently may find applications for many variants of Pattern Mining and outside of Pattern Mining, in particular in problems already tackled by sampling-based algorithms and concepts from Statistical Learning Theory; examples are the analysis of large networks (Riondato and Upfal, 2018; Borassi and Natale, 2019) and altered pathways in cancer (Vandin et al., 2016). We believe it is likely that the improvements we observed in Chapter 6 from using the n-MCERA w.r.t. other worst-case and distribution-free upper bounds would

transfer to other settings. In addition, we remark that in Chapter 4 we denoted an interesting connection between the probabilistic guarantees of uniform convergence with error rates of interest in Multiple Hypothesis Testing, such as the $FWER$; whether techniques based on the n-MCERA apply effectively to this fundamental problem is an interesting future research direction.

In Chapter 7 we studied the self-bounding properties of the n-MCERA, and have shown that they allow to derive novel sharper concentration bounds w.r.t. its expectation. Obtaining tight error rates on the n-MCERA is of central importance to obtain tight probabilistic upper bounds on the Rademacher Averages and, therefore, uniform deviation bounds to the maximum deviation between empirical means and their expectations of sets of functions. We believe the novel bounds we proved in Chapter 7 to the convergence of the n-MCERA and to the Supremum Deviations should find direct application in the Pattern Mining problems we tackled with MCRapper in Chapter 6, and, as we discussed, possibly in other settings. Most importantly, newly derived results on concentration bounds for general self-bounding functions would be applicable to our settings, due to the self-bounding properties we proved on the n-MCERA and the Supremum Deviations. Then, while in this work we focused on deriving concentration results valid with high probability in finite samples, another interesting direction is to combine the self-bounding properties we proved with asymptotical concentration results, such as the Central Limit Theorem for martingales (Hall and Heyde, 2014). In fact, De Stefani and Upfal (2019) have shown how to apply this asymptotic result to bound the Supremum Deviation from the n-MCERA; as they discuss, in many applications asymptotic bounds may be preferred as they may be sharper than their finite-sample counterparts, in particular when the size of the analysed data is sufficiently large and the convergence to the normal distribution is reasonably accurate. Therefore, an interesting question is whether the self-bounding properties we proved in this work enable a sharper application of the Central Limit Theorem for martingales.

Another fundamental and extremely interesting future reseach direction is to consider the concentration of unbounded functions (Kontorovich, 2014; Mendelson, 2014; Cortes et al., 2019; Grünwald and Mehta, 2020), of great interest in many applications.

Bibliography

Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22:207–216.

Agrawal, R. and Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. In *Proc. 20th Int. Conf. Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering,*, ICDE'95, pages 3–14. IEEE.

Ahmed, N. K., Neville, J., Rossi, R. A., and N., D. (2015). Efficient Graphlet Counting for Large Networks. In *2015 IEEE International Conference on Data Mining*, pages 1–10.

Atzmueller, M. (2015). Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1):35–49.

Audano, P. and Vannberg, F. (2014). KAnalyze: a fast versatile pipelined K-mer toolkit. *Bioinformatics*, 30(14):2070–2072.

Barnard, G. A. (1945). A new test for 2×2 tables. *Nature*, 156:177.

Bartlett, P. L., Boucheron, S., and Lugosi, G. (2002). Model selection and error estimation. *Machine Learning*, 48(1-3):85–113.

Bartlett, P. L., Bousquet, O., Mendelson, S., et al. (2005). Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537.

Bartlett, P. L. and Mendelson, S. (2002). Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482.

Bay, S. D. and Pazzani, M. J. (2001). Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246.

Bayardo Jr, R. J. (1998). Efficiently mining long patterns from databases. *ACM Sigmod Record*, 27(2):85–93.

Belfodil, A., Belfodil, A., Bendimerad, A., Lamarre, P., Robardet, C., Kaytoue, M., and Plantevit, M. (2019). FSSD-A Fast and Efficient Algorithm for Subgroup Set Discovery. In *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 91–99. IEEE.

Belfodil, A., Belfodil, A., and Kaytoue, M. (2018). Anytime subgroup discovery in numerical domains with guarantees. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 500–516. Springer.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300.

Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *The annals of statistics*, 29(4):1165–1188.

Benoit, G., Peterlongo, P., Mariadassou, M., Drezen, E., Schbath, S., Lavenier, D., and Lemaitre, C. (2016). Multiple comparative metagenomics using multiset k-mer counting. *PeerJ Computer Science*, 2:e94.

Berger, R. (1994). Power comparison of exact unconditional tests for comparing two binomial proportions. *Institute of Statistics Mimeo Series*.

Berger, R. L. (1996). More powerful tests from confidence interval p values. *The American Statistician*, 50(4):314–318.

Bhatia, R. and Davis, C. (2000). A better bound on the variance. *The American Mathematical Monthly*, 107(4):353–357.

Boley, M., Lucchese, C., Paurat, D., and Gärtner, T. (2011). Direct local pattern sampling by efficient two-step random procedures. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11*.

Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62.

Borassi, M. and Natale, E. (2019). KADABRA is an adaptive algorithm for betweenness via random approximation. *Journal of Experimental Algorithmics (JEA)*, 24(1):1–35.

Boschloo, R. D. (1970). Raised conditional level of significance for the 2× 2-table when testing the equality of two probabilities. *Statistica Neerlandica*, 24(1):1–9.

Boucheron, S., Bousquet, O., and Lugosi, G. (2005). Theory of classification: A survey of some recent advances. *ESAIM: probability and statistics*, 9:323–375.

Boucheron, S., Lugosi, G., and Massart, P. (2000). A sharp concentration inequality with applications. *Random Structures & Algorithms*, 16(3):277–292.

Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press.

Boucheron, S., Lugosi, G., Massart, P., et al. (2009). On concentration of self-bounding functions. *Electronic Journal of Probability*, 14:1884–1899.

Bousquet, O. (2002). A Bennett concentration inequality and its application to suprema of empirical processes. *Comptes Rendus Mathematique*, 334(6):495–500.

Bousquet, O. (2003). Concentration inequalities for sub-additive functions using the entropy method. In *Stochastic inequalities and applications*, pages 213–247. Springer.

Bousquet, O., Koltchinskii, V., and Panchenko, D. (2002). Some local measures of complexity of convex hulls and generalization bounds. In *International Conference on Computational Learning Theory*, pages 59–73. Springer.

Břinda, K., Sykulski, M., and Kucherov, G. (2015). Spaced seeds improve k-mer-based metagenomic classification. *Bioinformatics*, 31(22):3584–3592.

Brown, C. T., Howe, A., Zhang, Q., Pyrkosz, A. B., and Brom, T. H. (2012). A reference-free algorithm for computational normalization of shotgun sequencing data. *arXiv preprint arXiv:1203.4802*.

Chakaravarthy, V. T., Pandit, V., and Sabharwal, Y. (2009). Analysis of sampling techniques for association rule mining. In *Proc. 12th Int. Conf. Database Theory*, ICDT '09, pages 276–283, New York, NY, USA. ACM.

Chikhi, R. and Medvedev, P. (2013). Informed and automated k-mer size selection for genome assembly. *Bioinformatics*, 30(1):31–37.

Choi, L., Blume, J. D., and Dupont, W. D. (2015). Elucidating the foundations of statistical inference with 2×2 tables. *PloS one*, 10(4):e0121263.

Cortes, C., Greenberg, S., and Mohri, M. (2019). Relative deviation learning bounds and generalization with unbounded loss functions. *Annals of Mathematics and Artificial Intelligence*, 85(1):45–70.

Danovaro, R., Canals, M., Tangherlini, M., Dell'Anno, A., Gambi, C., Lastras, G., Amblas, D., Sanchez-Vidal, A., Frigola, J., Calafat, A. M., et al. (2017). A submarine volcanic eruption leads to a novel microbial habitat. *Nature ecology & evolution*, 1(6):0144.

de Lima, A. M., da Silva, M. V., and Vignatti, A. L. (2020). Estimating the Percolation Centrality of Large Networks through Pseudo-dimension Theory. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1839–1847.

De Stefani, L. and Upfal, E. (2019). A Rademacher Complexity Based Method for Controlling Power and Confidence Level in Adaptive Statistical Analysis. In *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 71–80.

Dickson, L. B., Jiolle, D., Minard, G., Moltini-Conclois, I., Volant, S., Ghozlane, A., Bouchier, C., Ayala, D., Paupy, C., Moro, C. V., et al. (2017). Carryover effects of larval exposure to different environmental bacteria drive adult trait variation in a mosquito vector. *Science advances*, 3(8):e1700585.

Dong, G. and Bailey, J. (2012). *Contrast data mining: concepts, algorithms, and applications*. CRC Press.

Dzyuba, V., van Leeuwen, M., and De Raedt, L. (2017). Flexible constrained sampling with guarantees for pattern mining. *Data Mining and Knowledge Discovery*, 31(5):1266–1293.

Fischer, J. and Vreeken, J. (2020). Discovering Succinct Pattern Sets Expressing Co-Occurrence and Mutual Exclusivity. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 813–823.

Fisher, R. A. (1922). On the interpretation of χ 2 from contingency tables, and the calculation of P. *Journal of the Royal Statistical Society*, 85(1):87–94.

Fournier-Viger, P., Chun-Wei Lin, J., Truong-Chi, T., and Nkambou, R. (2019). A Survey of High Utility Itemset Mining. In *High-Utility Pattern Mining*. Springer International Publishing.

Gart, J. J., Chu, K. C., and Tarone, R. E. (1979). Statistical issues in interpretation of chronic bioassay tests for carcinogenicity. *Journal of the National Cancer Institute*, 62(4):957–974.

Girotto, S., Pizzi, C., and Comin, M. (2016). MetaProb: accurate metagenomic reads binning based on probabilistic sequence signatures. *Bioinformatics*, 32(17):i567–i575.

Grünwald, P. D. and Mehta, N. A. (2020). Fast Rates for General Unbounded Loss Functions: From ERM to Generalized Bayes. *Journal of Machine Learning Research*, 21(56):1–80.

Hall, P. and Heyde, C. C. (2014). *Martingale limit theory and its application*. Academic press.

Hämäläinen, W. (2012). Kingfisher: an efficient algorithm for searching for both positive and negative dependency rules with statistical significance measures. *Knowledge and information systems*, 32(2):383–414.

Hämäläinen, W. (2016). New upper bounds for tight and fast approximation of Fisher's exact test in dependency rule mining. *Computational Statistics & Data Analysis*, 93:469–482.

Hämäläinen, W. and Webb, G. I. (2019). A tutorial on statistically sound pattern discovery. *Data Mining and Knowledge Discovery*, 33(2):325–377.

Han, J., Cheng, H., Xin, D., and Yan, X. (2007). Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15:55–86.

Han, J., Pei, J., and Yin, Y. (2000). Mining Frequent Patterns without Candidate Generation. In Chen, W., Naughton, J. F., and Bernstein, P. A., editors, *SIGMOD Conf.*, pages 1–12. ACM.

Han, J., Wang, J., Lu, Y., and Tzvetkov, P. (2002). Mining top-k frequent closed patterns without minimum support. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 211–218. IEEE.

Herrera, F., Carmona, C. J., González, P., and Del Jesus, M. J. (2011). An overview on subgroup discovery: foundations and applications. *Knowledge and information systems*, 29(3):495–525.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70.

Hrytsenko, Y., Daniels, N. M., and Schwartz, R. S. (2018). Efficient Distance Calculations Between Genomes Using Mathematical Approximation. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 546–546. ACM.

Jiang, C., Coenen, F., and Zito, M. (2013). A survey of frequent subgraph mining algorithms. *Knowledge Engineering Review*, 28(1):75–105.

Kalofolias, J., Boley, M., and Vreeken, J. (2017). Efficiently discovering locally exceptional yet globally representative subgroups. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 197–206. IEEE.

Kelley, D. R., Schatz, M. C., and Salzberg, S. L. (2010). Quake: quality-aware detection and correction of sequencing errors. *Genome biology*, 11(11):R116.

Kirsch, A., Mitzenmacher, M., Pietracaprina, A., Pucci, G., Upfal, E., and Vandin, F. (2012). An efficient rigorous approach for identifying statistically significant frequent itemsets. *Journal of the ACM (JACM)*, 59(3):12.

Klösgen, W. (1992). Problems for knowledge discovery in databases and their treatment in the Statistics Interpreter Explora. *International Journal of Intelligent Systems*, 7:649–673.

Knobbe, A. J. and Ho, E. K. (2006). Maximally informative k-itemsets and their efficient discovery. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 237–244.

Kokot, M., Długosz, M., and Deorowicz, S. (2017). KMC 3: counting and manipulating k-mer statistics. *Bioinformatics*, 33(17):2759–2761.

Koltchinskii, V. (2006). Local Rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2593–2656.

Koltchinskii, V. (2011). *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole d'Eté de Probabilités de Saint-Flour XXXVIII-2008*, volume 2033. Springer Science & Business Media.

Koltchinskii, V. and Panchenko, D. (2000). Rademacher processes and bounding the risk of function learning. In *High dimensional probability II*, pages 443–457. Springer.

Komiyama, J., Ishihata, M., Arimura, H., Nishibayashi, T., and Minato, S.-i. (2017). Statistical Emerging Pattern Mining with Multiple Testing Correction. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 897–906. ACM.

Kontorovich, A. (2014). Concentration in unbounded metric spaces and algorithmic stability. In *International Conference on Machine Learning*, pages 28–36.

Kurtz, S., Narechania, A., Stein, J. C., and Ware, D. (2008). A new method to compute K-mer frequencies and its application to annotate large repetitive plant genomes. *BMC genomics*, 9(1):517.

Lehmann, E. L. and Romano, J. P. (2012). Generalizations of the familywise error rate. In *Selected Works of EL Lehmann*, pages 719–735. Springer.

Lentz, W. J. (1976). Generating Bessel functions in Mie scattering calculations using continued fractions. *Applied Optics*, 15(3):668–671.

Li, A. and Barber, R. F. (2019). Multiple testing with the structure-adaptive Benjamini–Hochberg algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(1):45–74.

Li, X. and Waterman, M. S. (2003). Estimating the repeat structure and length of DNA sequences using ℓ-tuples. *Genome research*, 13(8):1916–1922.

Llinares-López, F., Papaxanthos, L., Bodenham, D., Roqueiro, D., Investigators, C., and Borgwardt, K. (2017). Genome-wide genetic heterogeneity discovery with categorical covariates. *Bioinformatics*, 33(12):1820–1828.

Llinares-López, F., Papaxanthos, L., Roqueiro, D., Bodenham, D., and Borgwardt, K. (2019). CASMAP: detection of statistically significant combinations of SNPs in association mapping. *Bioinformatics*, 35(15):2680–2682.

Llinares-López, F., Sugiyama, M., Papaxanthos, L., and Borgwardt, K. (2015). Fast and memory-efficient significant pattern mining via permutation testing. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 725–734. ACM.

Löffler, M. and Phillips, J. M. (2009). Shape fitting on point sets with probability distributions. In *European Symposium on Algorithms*, pages 313–324. Springer.

Mantel, N. (1980). A biometrics invited paper. assessing laboratory evidence for neoplastic activity. *Biometrics*, pages 381–399.

Marçais, G. and Kingsford, C. (2011). A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6):764–770.

Massart, P. (2000). Some applications of concentration inequalities to statistics. *Annales de la Faculté des sciences de Toulouse: Mathématiques*, 9(2):245–303.

McDiarmid, C. (1989). On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188.

Mehta, C. R. and Senchaudhuri, P. (2003). Conditional versus unconditional exact tests for comparing two binomials. *Cytel Software Corporation*, 675:1–5.

Meinshausen, N., Maathuis, M. H., Bühlmann, P., et al. (2011). Asymptotic optimality of the Westfall–Young permutation procedure for multiple testing under dependence. *The Annals of Statistics*, 39(6):3369–3391.

Melsted, P. and Halldórsson, B. V. (2014). KmerStream: streaming algorithms for k-mer abundance estimation. *Bioinformatics*, 30(24):3541–3547.

Melsted, P. and Pritchard, J. K. (2011). Efficient counting of k-mers in DNA sequences using a bloom filter. *BMC bioinformatics*, 12(1):333.

Mendelson, S. (2002). Improving the sample complexity using global data. *IEEE transactions on Information Theory*, 48(7):1977–1991.

Mendelson, S. (2014). Learning without concentration. In *Conference on Learning Theory*, pages 25–39.

Minato, S.-i., Uno, T., Tsuda, K., Terada, A., and Sese, J. (2014). A fast method of statistical assessment for combinatorial hypotheses based on frequent itemset enumeration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 422–436. Springer.

Mitzenmacher, M. and Upfal, E. (2017). *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press.

Mohamadi, H., Khan, H., and Birol, I. (2017). ntCard: a streaming algorithm for cardinality estimation in genomics data. *Bioinformatics*, 33(9):1324–1330.

Nijssen, S. and Kok, J. N. (2004). A quickstart in frequent structure mining can make a difference. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 647–652. ACM.

Nijssen, S. and Kok, J. N. (2006). Frequent subgraph miners: runtimes don't say everything. *MLG 2006*, page 173.

Ondov, B. D., Treangen, T. J., Melsted, P., Mallonee, A. B., Bergman, N. H., Koren, S., and Phillippy, A. M. (2016). Mash: fast genome and metagenome distance estimation using MinHash. *Genome biology*, 17(1):132.

Oneto, L., Ghio, A., Anguita, D., and Ridella, S. (2013). An improved analysis of the Rademacher data-dependent bound using its self bounding property. *Neural Networks*, 44:107–111.

Oneto, L., Ghio, A., Ridella, S., and Anguita, D. (2016). Global rademacher complexity bounds: From slow to fast convergence rates. *Neural Processing Letters*, 43(2):567–602.

Pandey, P., Bender, M. A., Johnson, R., and Patro, R. (2017). Squeakr: an exact and approximate k-mer counting system. *Bioinformatics*.

Papaxanthos, L., Llinares-López, F., Bodenham, D., and Borgwardt, K. (2016). Finding significant combinations of features in the presence of categorical covariates. In *Advances in Neural Information Processing Systems*, pages 2279–2287.

Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (1999). Discovering frequent closed itemsets for association rules. In *International Conference on Database Theory*, pages 398–416. Springer.

Patro, R., Mount, S. M., and Kingsford, C. (2014). Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nature biotechnology*, 32(5):462.

Pearson, K. (1900). X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175.

Pellegrina, L. (2020). Sharper convergence bounds of Monte Carlo Rademacher Averages through Self-Bounding functions. *arXiv preprint arXiv:2010.12103*.

Pellegrina, L., Cousins, C., Vandin, F., and Riondato, M. (2020a). MCRapper: Monte-Carlo Rademacher Averages for Poset Families and Approximate Pattern Mining. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2165–2174.

Pellegrina, L., Pizzi, C., and Vandin, F. (2019a). Fast Approximation of Frequent k-mers and Applications to Metagenomics. In *International Conference on Research in Computational Molecular Biology*, pages 208–226. Springer.

Pellegrina, L., Pizzi, C., and Vandin, F. (2020b). Fast Approximation of Frequent k-mers and Applications to Metagenomics. *Journal of Computational Biology*, 27(4):534–549.

Pellegrina, L., Riondato, M., and Vandin, F. (2019b). Hypothesis Testing and Statistically-sound Pattern Mining. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3215–3216. ACM.

Pellegrina, L., Riondato, M., and Vandin, F. (2019c). SPuManTE: Significant Pattern Mining with Unconditional Testing. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 1528–1538, New York, NY, USA. ACM.

Pellegrina, L. and Vandin, F. (2018). Efficient mining of the most significant patterns with permutation testing. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2070–2079. ACM.

Pellegrina, L. and Vandin, F. (2020). Efficient mining of the most significant patterns with permutation testing. *Data Mining and Knowledge Discovery*, 34(4):1201–1234.

Pevzner, P. A., Tang, H., and Waterman, M. S. (2001). An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences*, 98(17):9748–9753.

Pietracaprina, A., Riondato, M., Upfal, E., and Vandin, F. (2010). Mining top-K frequent itemsets through progressive sampling. *Data Mining Knowl. Disc.*, 21(2):310–326.

Pietracaprina, A. and Vandin, F. (2007). Efficient Incremental Mining of Top-K Frequent Closed Itemsets. In *Discovery Science*, volume 4755 of *Lecture Notes in Computer Science*, pages 275–280. Springer Berlin Heidelberg.

Pollard, D. (1984). *Convergence of stochastic processes*. Springer-Verlag.

Popoviciu, T. (1935). Sur les équations algébriques ayant toutes leurs racines réelles. *Mathematica*, 9:129–145.

Relator, R. T., Terada, A., and Sese, J. (2018). Identifying statistically significant combinatorial markers for survival analysis. *BMC medical genomics*, 11(2):45–55.

Riondato, M. and Kornaropoulos, E. M. (2016). Fast approximation of betweenness centrality through sampling. *Data Mining and Knowledge Discovery*, 30(2):438–475.

Riondato, M. and Upfal, E. (2014). Efficient Discovery of Association Rules and Frequent Itemsets through Sampling with Tight Performance Guarantees. *ACM Trans. Knowl. Disc. from Data*, 8(4):20.

Riondato, M. and Upfal, E. (2015). Mining frequent itemsets through progressive sampling with Rademacher averages. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1005–1014. ACM, ACM.

Riondato, M. and Upfal, E. (2018). ABRA: Approximating Betweenness Centrality in Static and Dynamic Graphs with Rademacher Averages. *ACM Trans. Knowl. Disc. from Data*, 12(5):61.

Riondato, M. and Vandin, F. (2014). Finding the true frequent itemsets. In *Proceedings of the 2014 SIAM international conference on data mining*, pages 497–505. SIAM.

Riondato, M. and Vandin, F. (2018). MiSoSouP: Mining Interesting Subgroups with Sampling and Pseudodimension. In *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Disc. and Data Mining*, KDD '18, pages 2130–2139. ACM.

Rizk, G., Lavenier, D., and Chikhi, R. (2013). DSK: k-mer counting with very low memory usage. *Bioinformatics*, 29(5):652–653.

Romano, J. P., Shaikh, A. M., et al. (2006a). On stepdown control of the false discovery proportion. In *Optimality*, pages 33–50. Institute of Mathematical Statistics.

Romano, J. P., Shaikh, A. M., et al. (2006b). Stepup procedures for control of generalizations of the familywise error rate. *The Annals of Statistics*, 34(4):1850–1873.

Romano, J. P. and Wolf, M. (2005). Exact and approximate stepdown methods for multiple hypothesis testing. *Journal of the American Statistical Association*, 100(469):94–108.

Roy, R. S., Bhattacharya, D., and Schliep, A. (2014). Turtle: Identifying frequent k-mers with cache-efficient algorithms. *Bioinformatics*, 30(14):1950–1957.

Salmela, L., Walve, R., Rivals, E., and Ukkonen, E. (2016). Accurate self-correction of errors in long reads using de Bruijn graphs. *Bioinformatics*, 33(6):799–806.

Santoro, D., Pellegrina, L., and Vandin, F. (2021). SPRISS: Approximating Frequent k-mers by Sampling Reads, and Applications. *arXiv preprint arXiv:2101.07117 (to appear at RECOMB 2021)*.

Santoro, D., Tonon, A., and Vandin, F. (2020). Mining Sequential Patterns with VC-Dimension and Rademacher Complexity. *Algorithms*, 13(5):123.

Sarpe, I. and Vandin, F. (2021). PRESTO: Simple and Scalable Sampling Techniques for the Rigorous Approximation of Temporal Motif Counts. *arXiv preprint arXiv:2101.07152 (to appear at SDM 2021)*.

Servan-Schreiber, S., Riondato, M., and Zgraggen, E. (2018a). ProSecCo: Progressive Sequence Mining with Convergence Guarantees. In *Proceedings of the 18th IEEE International Conference on Data Mining*, pages 417–426.

Servan-Schreiber, S., Riondato, M., and Zgraggen, E. (2018b). ProSecCo: Progressive sequence mining with convergence guarantees. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 417–426. IEEE.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

Sims, G. E., Jun, S.-R., Wu, G. A., and Kim, S.-H. (2009). Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. *Proceedings of the National Academy of Sciences*, 106(8):2677–2682.

Sivadasan, N., Srinivasan, R., and Goyal, K. (2016). Kmerlight: fast and accurate k-mer abundance estimation. *arXiv preprint arXiv:1609.05626*.

Solomon, B. and Kingsford, C. (2016). Fast search of thousands of short-read sequencing experiments. *Nature biotechnology*, 34(3):300.

Sugiyama, M. and Borgwardt, K. M. (2019). Finding Statistically Significant Interactions between Continuous Features. In *IJCAI*, pages 3490–3498.

Sugiyama, M., Llinares-López, F., Kasenburg, N., and Borgwardt, K. M. (2015). Significant subgraph mining with multiple testing correction. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 37–45. SIAM.

Talagrand, M. (1994). Sharper bounds for Gaussian and empirical processes. *The Annals of Probability*, 22(1):28–76.

Talagrand, M. (1995). Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l'Institut des Hautes Etudes Scientifiques*, 81(1):73–205.

Tarone, R. (1990). A modified Bonferroni method for discrete data. *Biometrics*, pages 515–522.

Terada, A., Kim, H., and Sese, J. (2015). High-speed Westfall-Young permutation procedure for genome-wide association studies. In *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 17–26. ACM.

Terada, A., Okada-Hatakeyama, M., Tsuda, K., and Sese, J. (2013a). Statistical significance of combinatorial regulations. *Proceedings of the National Academy of Sciences*, 110(32):12996–13001.

Terada, A., Tsuda, K., et al. (2016). Significant pattern mining with confounding variables. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 277–289. Springer.

Terada, A., Tsuda, K., and Sese, J. (2013b). Fast Westfall-Young permutation procedure for combinatorial regulation discovery. In *2013 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 153–158. IEEE.

Toivonen, H. (1996). Sampling Large Databases for Association Rules. In *Proc. 22nd Int. Conf. Very Large Data Bases*, VLDB '96, pages 134–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Tonon, A. and Vandin, F. (2019). Permutation Strategies for Mining Significant Sequential Patterns. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1330–1335. IEEE.

Uno, T., Kiyomi, M., and Arimura, H. (2005). LCM ver. 3: collaboration of array, bitmap and prefix tree for frequent itemset mining. In *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, pages 77–86. ACM.

van der Laan, M. J., Dudoit, S., and Pollard, K. S. (2004). Augmentation procedures for control of the generalized family-wise error rate and tail probabilities for the proportion of false positives. *Statistical applications in genetics and molecular biology*, 3(1):1–25.

van Leeuwen, M., De Bie, T., Spyropoulou, E., and Mesnage, C. (2016). Subjective interestingness of subgraph patterns. *Machine Learning*, 105(1):41–75.

Van Leeuwen, M. and Knobbe, A. (2012). Diverse subgroup set discovery. *Data Mining and Knowledge Discovery*, 25(2):208–242.

Vandin, F., Papoutsaki, A., Raphael, B. J., and Upfal, E. (2015). Accurate computation of survival statistics in genome-wide studies. *PLoS computational biology*, 11(5):e1004071.

Vandin, F., Raphael, B. J., and Upfal, E. (2016). On the sample complexity of cancer pathways identification. *Journal of Computational Biology*, 23(1):30–41.

Vapnik, V. N. (1998). *Statistical learning theory*. Wiley.

Vapnik, V. N. and Chervonenkis, A. Y. (1971). On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Theory of Probability & Its Applications*, 16(2):264.

Vreeken, J., Van Leeuwen, M., and Siebes, A. (2011). Krimp: mining itemsets that compress. *Data Mining and Knowledge Discovery*, 23(1):169–214.

Webb, G. I. (2006). Discovering significant rules. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 434–443. ACM.

Webb, G. I. (2007). Discovering significant patterns. *Machine learning*, 68(1):1–33.

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

Westfall, P. H. and Troendle, J. F. (2008). Multiple testing with minimal assumptions. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, 50(5):745–755.

Westfall, P. H. and Young, S. S. (1993). Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment.

Wood, D. E. and Salzberg, S. L. (2014). Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome biology*, 15(3):R46.

Wörlein, M., Meinl, T., Fischer, I., and Philippsen, M. (2005). A quantitative comparison of the subgraph miners MoFa, gSpan, FFSM, and Gaston. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 392–403. Springer.

Yates, F. (1984). Tests of significance for 2× 2 contingency tables. *Journal of the Royal Statistical Society: Series A (General)*, 147(3):426–449.

Zandolin, D. and Pietracaprina, A. (2003). Mining frequent itemsets using patricia tries. In *Proceedings of FIMI03*, volume 90.

Zerbino, D. R. and Birney, E. (2008). Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome research*, 18(5):821–829.

Zhang, Q., Pell, J., Canino-Koning, R., Howe, A. C., and Brown, C. T. (2014). These are not the k-mers you are looking for: efficient online k-mer counting using a probabilistic data structure. *PloS one*, 9(7):e101271.

Zhang, Z. and Wang, W. (2014). RNA-Skim: a rapid method for RNA-Seq quantification at transcript level. *Bioinformatics*, 30(12):i283–i292.