algorithms

Article

Mining Sequential Patterns with VC-Dimension and Rademacher Complexity

Diego Santoro †, **Andrea Tonon** † and **Fabio Vandin** *

Department of Information Engineering, University of Padova, 35131 Padova, Italy;
diego.santoro@dei.unipd.it (D.S.); andrea.tonon@dei.unipd.it (A.T.)
* Correspondence: fabio.vandin@unipd.it
† These authors contributed equally to this work.

Received: 10 April 2020; Accepted: 14 May 2020; Published: 18 May 2020

Abstract: Sequential pattern mining is a fundamental data mining task with application in several domains. We study two variants of this task—the first is the extraction of frequent sequential patterns, whose frequency in a dataset of sequential transactions is higher than a user-provided threshold; the second is the mining of true frequent sequential patterns, which appear with probability above a user-defined threshold in transactions drawn from the generative process underlying the data. We present the first sampling-based algorithm to mine, with high confidence, a rigorous approximation of the frequent sequential patterns from massive datasets. We also present the first algorithms to mine approximations of the true frequent sequential patterns with rigorous guarantees on the quality of the output. Our algorithms are based on novel applications of Vapnik-Chervonenkis dimension and Rademacher complexity, advanced tools from statistical learning theory, to sequential pattern mining. Our extensive experimental evaluation shows that our algorithms provide high-quality approximations for both problems we consider.

Keywords: data mining; sequential patterns; sampling; VC-dimension; Rademacher complexity; statistical learning

1. Introduction

Sequential pattern mining [1] is a fundamental task in data mining and knowledge discovery, with applications in several fields, from recommender systems and e-commerce to biology and medicine. In its original formulation, sequential pattern mining requires to identify all *frequent sequential patterns*, that is, sequences of itemsets that appear in a fraction at least θ of all the transactions in a transactional dataset, where each transaction is a sequence of itemsets. The threshold θ is a user-specified parameter and its choice must be, at least in part, be informed by domain knowledge. In general, sequential patterns describe sequences of events or actions that are useful for predictions in many scenarios.

Several exact methods have been proposed to find frequent sequential patterns. However, the exact solution of the problem requires processing the entire dataset at least once, and often multiple times. For large, modern sized datasets, this may be infeasible. A natural solution to reduce the computation is to use *sampling* to obtain a small random portion (*sample*) of the dataset, and perform the mining process only on the sample. It is easy to see that by analyzing only a sample of the data the problem cannot be solved exactly, and one has to rely on the approximation provided by the results of the mining task on the sample. Therefore, the main challenge in using sampling is on computing a sample size such that the frequency of the sequential patterns in the sample is close to the frequency that would be obtained from the analysis on the whole dataset. Relating the two quantities using standard techniques (e.g., Hoeffding inequality and union bounds) does not provide useful results, that is, small sample sizes. In fact, such procedures require the knowledge of the number of all the sequential patterns in the dataset, which is impractical to compute in a reasonable time. So, one has to resort to loose upper bounds that usually result in sample sizes that are larger than the whole dataset. Recently, tools from statistical learning (e.g.,Vapnik-Chervonenkis dimension [2] and Rademacher complexity [3]) have been successfully used in frequent itemsets mining [4,5], a frequent pattern mining task where transactions are collections of items, showing that accurate and rigorous approximations can be obtained from small samples of the entire dataset. While sampling has previously been used in the context of sequential pattern mining (e.g., Reference [6]), to the best of our knowledge no sampling algorithm providing a rigorous approximation of the frequent sequential patterns has been proposed.

In several applications, the analysis of a dataset is performed to gain insight on the *underlying generative process* of the data. For example, in market basket analysis one is interested in gaining knowledge on the behaviour of all the customers, which can be modelled as a generative process from which the transactions in the dataset have been drawn. In such a scenario, one is not interested in sequential patterns that are frequent *in the dataset*, but in sequential patterns that are frequent *in the generative process*, that is, whose probability of appearing in a transaction generated from the process is above a threshold θ. Such patterns, called *true frequent patterns*, have been introduced by Reference [7], which provides a Vapnik-Chervonenkis (VC) dimension based approach to mine true frequent itemsets. While there is a relation between the probability that a pattern appears in a transaction generated from the process and its frequency in the dataset, one cannot simply look at patterns with frequency above θ in the dataset to find the ones with probability above θ in the process. Moreover, due to the stochastic nature of the data, one cannot identify the true frequent patterns with certainty, and approximations are to be sought. In such a scenario, relating the probability that a pattern appears in a transaction generated from the process with its frequency in the dataset using standard techniques is even more challenging. Hoeffding inequality and union bounds require to bound the number of all the possible sequential patterns that can be generated from the process. Such bound is infinite if one considers all possible sequential patterns (e.g., does not bound the pattern length). To the best of our knowledge, no method to mine *true frequent sequential patterns* has been proposed.

1.1. Our Contributions

In this work, we study two problems in sequential pattern mining—mining *frequent sequential patterns* and mining *true frequent sequential patterns*. We propose efficient algorithms for these problems, based on the concepts of VC-dimension and Rademacher complexity. In this regard, our contributions are:

- We define rigorous approximations of the set of frequent sequential patterns and the set of true frequent sequential patterns. In particular, for both sets we define two approximations: one with no *false negatives*, that is, containing all elements of the set; and one with no *false positives*, that is, without any element that is not in the set. Our approximations are defined in terms of a single parameter, which controls the accuracy of the approximation and is easily interpretable.
- We study the VC-dimension and the Rademacher complexity of sequential patterns, two advanced concepts from statistical learning theory that have been used in other mining contexts, and provide algorithms to efficiently compute upper bounds for both. In particular, we provide a simple, but still effective in practice, upper bound to the VC-dimension of sequential patterns by relaxing the upper bound previously defined in Reference [8]. We also provide the first efficiently computable upper bound to the Rademacher complexity of sequential patterns. We also show how to approximate the Rademacher complexity of sequential patterns.
- We introduce a new sampling-based algorithm to identify rigorous approximations of the frequent sequential patterns with probability $1 - \delta$, where δ is a confidence parameter set by the user. Our algorithm hinges on our novel bound on the VC-dimension of sequential patterns, and it allows to obtain a rigorous approximation of the frequent sequential patterns by mining only a fraction of the whole dataset.

- We introduce efficient algorithms to obtain rigorous approximations of the true frequent sequential patterns with probability $1 - \delta$, where δ is a confidence parameter set by the user. Our algorithms use the novel bounds on the VC-dimension and on Rademacher complexity that we have derived, and they allow to obtain accurate approximations of the true frequent sequential patterns, where the accuracy depends on the size of the available data.
- We perform an extensive experimental evaluation analyzing several sequential datasets, showing that our algorithms provide high-quality approximations, even better than guaranteed by their theoretical analysis, for both tasks we consider.

1.2. Related Work

Since the introduction of the frequent sequential pattern mining problem [1], a number of exact algorithms has been proposed for this task, ranging from multi-pass algorithms using the anti-monotonicity property of the frequency function [9], to prefix-based approaches [10], to works focusing on the closed frequent sequences [11].

The use of sampling to reduce the amount of data for the mining process while obtaining rigorous approximations of the collection of interesting patterns has been successfully applied in many mining tasks. Raïssi and Poncelet [6] provided a theoretical bound on the sample size for a single sequential pattern in a static dataset using Hoeffding concentration inequalities, and they introduced a sampling approach to build a dynamic sample in a streaming scenario using a biased reservoir sampling. Our work is heavily inspired by the work of Riondato and Upfal [4,5], which introduced advanced statistical learning techniques for the task of frequent itemsets and association rules mining. In particular, in Reference [4] they employed the concept of VC-dimension to derive a bound on the sample size needed to obtain an approximation of the frequent itemsets and association rules from a dataset, while in Reference [5] they proposed a progressive sampling approach based on an efficiently computable upper bound on the Rademacher complexity of itemsets. VC-dimension has also been used to approximate frequent substrings in collections of strings [12], and the related concept of pseudo-dimension has been used to mine interesting subgroups [13]. Rademacher complexity has also been used in graph mining [14–16], to design random sampling approaches for estimating betweenness centralities in graphs [17].

Other works have studied the problem of approximating frequent sequential patterns using approaches other than sampling. In Reference [18], the dataset is processed in blocks with a streaming algorithm, but the intermediate sequential patterns returned may miss many frequent sequential patterns. More recently, Reference [8] introduced an algorithm to process the datasets in blocks using a variable, data-dependent frequency threshold, based on an upper bound to the empirical VC-dimension, to mine each block. Reference [8] defines an approximation for frequent sequential patterns that is one of the definitions we consider in this work. The intermediate results obtained after analyzing each block have probabilistic approximation guarantees, and after analyzing all blocks the output is the exact collection of frequent sequential patterns. While these works, in particular Reference [8], are related to our contributions, they do not provide sampling algorithms for sequential pattern mining.

To the best of our knowledge, Reference [7] is the only work that considers the extraction of frequent patterns w.r.t. an underlying generative process, based on the concept of empirical VC-dimension of itemsets. While we use the general framework introduced by Reference [7], the solution proposed by Reference [7] requires to solve an optimization problem that is tailored to itemsets and, thus, not applicable to sequential patterns; in addition, computing the solution of such problem could be relatively expensive. Reference [19] considers the problem of mining significant patterns under a similar framework, making more realistic assumptions on the underlying generative process compared to commonly used tests (e.g., Fisher's exact test).

Several works have been proposed to identify statistically significant patterns where the significance is defined in terms of the comparison of patterns statistics. Few methods [20–22] have been proposed to mine statistically significant sequential patterns. These methods are orthogonal to our approach, which focuses on finding sequential patterns that are frequent with respect to (w.r.t.) an underlying generative distribution.

2. Preliminaries

We now provide the definitions and concepts used throughout the article. We start by introducing the task of sequential pattern mining and formally define the two problems which are the focus of this work: approximating the frequent sequential patterns and mining sequential patterns that are frequently generated from the underlying generative process. We then introduce two tools from statistical learning theory, that is, the VC-dimension and the Rademacher complexity, and the related concept of maximum deviation.

2.1. Sequential Pattern Mining

Let $\mathcal{I} = \{i_1, i_2, \ldots, i_h\}$ be a finite set of elements called *items*. $\mathcal{I}$ is also called the *ground set*. An *itemset* P is a (non-empty) subset of $\mathcal{I}$, that is, $P \subseteq \mathcal{I}$. A *sequential pattern* $p = \langle P_1, P_2, \ldots, P_\ell \rangle$ is a *finite ordered sequence* of itemsets, with $P_i \subseteq \mathcal{I}, 1 \leq i \leq \ell$. A sequential pattern p is also called a *sequence*. The *length* $|p|$ of p is defined as the number of itemsets in p. The *item-length* $||p||$ of p is the sum of the sizes of the itemsets in p, that is,

$$||p|| = \sum_{i=1}^{|p|} |P_i|, \tag{1}$$

where $|P_i|$ is the number of items in itemset P_i. A sequence $a = \langle A_1, A_2, \ldots, A_m \rangle$ is a *subsequence* of another sequence $b = \langle B_1, B_2, \ldots, B_n \rangle$, denoted by $a \sqsubseteq b$, if and only if there exist integers $1 \leq i_1 < i_2 < \ldots < i_m \leq n$ such that $A_1 \subseteq B_{i_1}, A_2 \subseteq B_{i_2}, \ldots, A_m \subseteq B_{i_m}$. If a is a subsequence of b, then b is called a *super-sequence* of a, denoted by $b \sqsupseteq a$.

Let $\mathbb{U}$ denote the set of all the sequences which can be built with itemsets containing items from $\mathcal{I}$. A *dataset* $\mathcal{D}$ is a finite bag of *(sequential) transactions* where each transaction is a sequence from $\mathbb{U}$. A sequence p *belongs* to a transaction $\tau \in \mathcal{D}$ if and only if $p \sqsubseteq \tau$. For any sequence p, the *support set* $T_{\mathcal{D}}(p)$ of p in $\mathcal{D}$ is the set of transactions in $\mathcal{D}$ to which p belongs: $T_{\mathcal{D}}(p) = \{\tau \in D : p \sqsubseteq \tau\}$. The *support* $Supp_{\mathcal{D}}(p)$ of p in $\mathcal{D}$ is the cardinality of the set $T_{\mathcal{D}}(p)$, that is the number of transactions in $\mathcal{D}$ to which p belongs: $Supp_{\mathcal{D}}(p) = |T_{\mathcal{D}}(p)|$. Finally, the *frequency* $f_{\mathcal{D}}(p)$ of p in $\mathcal{D}$ is the *fraction* of transactions in $\mathcal{D}$ to which p belongs:

$$f_{\mathcal{D}}(p) = \frac{Supp_{\mathcal{D}}(p)}{|\mathcal{D}|}. \tag{2}$$

A sequence p is *closed* w.r.t. $\mathcal{D}$ if for each of its super-sequences $y \sqsupset p$ we have $f_{\mathcal{D}}(y) < f_{\mathcal{D}}(p)$, or, equivalently, none of its super-sequence has support equal to $f_{\mathcal{D}}(p)$. We denote the set of all closed sequences in $\mathcal{D}$ with $CS(\mathcal{D})$.

Example 1. *Consider the following dataset $\mathcal{D} = \{\tau_1, \tau_2, \tau_3, \tau_4\}$ as example:*

$$\begin{aligned}
\tau_1 &= \langle \{6,7\}, \{5\}, \{7\}, \{5\} \rangle \\
\tau_2 &= \langle \{1\}, \{2\}, \{6,7\}, \{5\} \rangle \\
\tau_3 &= \langle \{1,4\}, \{3\}, \{2\}, \{1,2,5,6\} \rangle \\
\tau_4 &= \langle \{1\}, \{2\}, \{6,7\}, \{5\} \rangle
\end{aligned}$$

The dataset above has 4 transactions. The first one, $\tau_1 = \langle \{6,7\}, \{5\}, \{7\}, \{5\} \rangle$, it is a sequence of length $|\tau_1| = 4$ and item-length $||\tau_1|| = 5$. The frequency $f_{\mathcal{D}}(\langle \{7\}, \{5\} \rangle)$ of $\langle \{7\}, \{5\} \rangle$ in $\mathcal{D}$, is 3/4, since it is contained in all transactions but τ_3. Note that the sequence $\langle \{7\}, \{5\} \rangle$ occurs three times as a subsequence of τ_1, but τ_1 contributes only once to the frequency of $\langle \{7\}, \{5\} \rangle$. The sequence $\langle \{7\}, \{6\}, \{5\} \rangle$ is not a subsequence of τ_1 because the order of the itemsets in the two sequences is not the same. Note that from the definitions above, *an item can only occur once in an itemset, but it can occur multiple times in different itemsets of the same sequence. Finally, the sequence $\langle\{6,7\},\{5\}\rangle$, whose frequency is 3/4, is a* closed *sequence, since its frequency is higher than the frequency of each of its super-sequences.*

Sections 2.1.1 and 2.1.2 formally define the two problems we are interested in.

2.1.1. Frequent Sequential Pattern Mining

Given a dataset $\mathcal{D}$ and a *minimum frequency threshold* $\theta \in (0,1]$, *frequent sequential pattern* (FSP) mining is the task of reporting the set $FSP(\mathcal{D},\theta)$ of all the sequences whose frequency in $\mathcal{D}$ is at least θ, and their frequencies:

$$FSP(\mathcal{D},\theta) = \{(p, f_{\mathcal{D}}(p)) : p \in \mathbb{U}, f_{\mathcal{D}}(p) \geq \theta\}. \tag{3}$$

In the first part of this work, we are interested in finding the set $FSP(\mathcal{D},\theta)$ by only mining a sample of the dataset $\mathcal{D}$. Note that given a sample of the dataset $\mathcal{D}$, one cannot guarantee to find the exact set $FSP(\mathcal{D},\theta)$ and has to resort to approximations of $FSP(\mathcal{D},\theta)$. Thus, we are interested in finding rigorous approximations of $FSP(\mathcal{D},\theta)$. In particular, we consider the approximation of $FSP(\mathcal{D},\theta)$ defined in Reference [8].

Definition 1. *Given $\varepsilon \in (0,1)$, an ε-approximation $\mathcal{C}$ of $FSP(\mathcal{D},\theta)$ is defined as a set of pairs (p, f_p):*

$$\mathcal{C} = \{(p, f_p) : p \in \mathbb{U}, f_p \in [0,1]\} \tag{4}$$

that has the following properties:

- *$\mathcal{C}$ contains a pair (p, f_p) for every $(p, f_{\mathcal{D}}(p)) \in FSP(\mathcal{D},\theta)$;*
- *$\mathcal{C}$ contains no pair (p, f_p) such that $f_{\mathcal{D}}(p) < \theta - \varepsilon$;*
- *for every $(p, f_p) \in \mathcal{C}$, it holds $|f_{\mathcal{D}}(p) - f_p| \leq \varepsilon/2$.*

(Note that while Reference [8] introduced the definition of ε-approximation of $FSP(\mathcal{D},\theta)$, it did not provide a sampling algorithm to find such approximation for a given $\varepsilon \in (0,1)$.)

Intuitively, the approximation $\mathcal{C}$ contains all the frequent sequential patterns that are in $FSP(\mathcal{D},\theta)$ (i.e., there are no *false negatives*) and no sequential pattern that has frequency in $\mathcal{D}$ much below θ. In addition, $\mathcal{C}$ provides a good approximation of the actual frequency of the sequential pattern in $\mathcal{D}$, within an error $\varepsilon/2$, arbitrarily small.

Depending on the application, one may be interested in a different approximation of $FSP(\mathcal{D},\theta)$, where all the sequential patterns in the approximation are frequent sequential patterns in the whole dataset.

Definition 2. *Given $\varepsilon \in (0,1)$, a false positives free (FPF) ε-approximation $\mathcal{F}$ of $FSP(\mathcal{D},\theta)$ is defined as a set of pairs (p, f_p):*

$$\mathcal{F} = \{(p, f_p) : p \in \mathbb{U}, f_p \in [0,1]\} \tag{5}$$

that has the following properties:

- *$\mathcal{F}$ contains no pair (p, f_p) such that $f_{\mathcal{D}}(p) < \theta$;*
- *$\mathcal{F}$ contains all the pairs (p, f_p) such that $f_{\mathcal{D}}(p) \geq \theta + \varepsilon$;*
- *for every $(p, f_p) \in \mathcal{F}$, it holds $|f_{\mathcal{D}}(p) - f_p| \leq \varepsilon/2$.*

The approximation $\mathcal{F}$ does not contain *false positives*, that is, sequences with $f_{\mathcal{D}}(p) < \theta$. In addition, it does not miss sequences with $f_{\mathcal{D}}(p) \geq \theta + \varepsilon$ and, similarly to the ε-approximation, we have that, for every pair in $\mathcal{F}$, it gives a good approximation of the actual frequency of the sequential patterns in $\mathcal{D}$, within an error $\varepsilon/2$, arbitrarily small.

2.1.2. True Frequent Sequential Pattern Mining

In several applications, the dataset $\mathcal{D}$ is a sample of transactions independently drawn from an unknown probability distribution π on $\mathbb{U}$. In such a scenario, the dataset $\mathcal{D}$ is a finite bag of $|\mathcal{D}|$ *independent identically distributed (i.i.d.)* samples from π. For any sequence $p \in \mathbb{U}$, the *real support set* $T(p)$ of p is the set of sequences in $\mathbb{U}$ to which p belongs: $T(p) = \{\tau \in \mathbb{U} : p \sqsubseteq \tau\}$. We define the *true frequency* $t_\pi(p)$ of p w.r.t. π as the probability that a transaction sampled from π contains p:

$$t_\pi(p) = \sum_{\tau \in T(p)} \pi(\tau). \tag{6}$$

In this scenario, the final goal of the data mining process on $\mathcal{D}$ is to gain a better understanding of the process generating the data, that is, of the distribution π, through the true frequencies t_π, which are unknown and only approximately reflected in the dataset $\mathcal{D}$. Therefore, we are interested in finding the sequential patterns with true frequency t_π at least θ for some $\theta \in (0,1]$. We call these sequential patterns the *true frequent sequential patterns* (TFSPs) and denote their set as:

$$TFSP(\pi,\theta) = \{(p, t_\pi(p)) : p \in \mathbb{U}, t_\pi(p) \geq \theta\}. \tag{7}$$

Note that, given a finite number of random samples from π (e.g., the dataset $\mathcal{D}$), it is not possible to find the exact set $TFSP(\pi,\theta)$, and one has to resort to approximations of $TFSP(\pi,\theta)$. Analogously to the two approximations defined for the FSPs, now we define two approximations of the TFSPs, depending on the application we are interested in: the first one that does not have false negatives, while the second one that does not contain false positives.

Definition 3. *Given $\mu \in (0,1)$, a μ-approximation $\mathcal{E}$ of $TFSP(\pi,\theta)$ is defined as a set of pairs (p, f_p):*

$$\mathcal{E} = \{(p, f_p) : p \in \mathbb{U}, f_p \in [0,1]\} \tag{8}$$

that has the following properties:

- *$\mathcal{E}$ contains a pair (p, f_p) for every $(p, t_\pi(p)) \in TFSP(\pi,\theta)$;*
- *$\mathcal{E}$ contains no pair (p, f_p) such that $t_\pi(p) < \theta - \mu$;*
- *for every $(p, f_p) \in \mathcal{E}$, it holds $|t_\pi(p) - f_p| \leq \mu/2$.*

Definition 4. *Given $\mu \in (0,1)$, a false positives free (FPF) μ-approximation $\mathcal{G}$ of $TFSP(\pi,\theta)$ is defined as a set of pairs (p, f_p):*

$$\mathcal{G} = \{(p, f_p) : p \in \mathbb{U}, f_p \in [0,1]\} \tag{9}$$

that has the following properties:

- *$\mathcal{G}$ contains no pair (p, f_p) such that $t_\pi(p) < \theta$;*
- *$\mathcal{G}$ contains all the pairs (p, f_p) such that $t_\pi(p) \geq \theta + \mu$;*
- *for every $(p, f_p) \in \mathcal{G}$, it holds $|t_\pi(p) - f_p| \leq \mu/2$.*

2.2. VC-Dimension

The Vapnik-Chervonenkis (VC) dimension [2,23] of a space of points is a measure of the complexity or expressiveness of a family of indicator functions, or, equivalently, of a family of subsets, defined on that space. A finite bound on the VC-dimension of a structure implies a bound of the number of random samples required to approximately learn that structure.

We define a *range space* as a pair $(X, \mathcal{R})$, where X is a finite or infinite set and $\mathcal{R}$, the *range set*, is a finite or infinite family of subsets of X. The members of X are called *points*, while the members of $\mathcal{R}$ are called *ranges*. Given $A \subseteq X$, we define the *projection* of $\mathcal{R}$ in A as $P_{\mathcal{R}}(A) = \{r \cap A \ : \ r \in \mathcal{R}\}$.

We define 2^A as the *power set* of A, that is the set of all the possible subsets of A, including the empty set $\emptyset$ and A itself. If $P_{\mathcal{R}}(A) = 2^A$, then A is said to be *shattered by* $\mathcal{R}$. The VC-dimension of a range space is the cardinality of the largest set shattered by the space.

Definition 5. *Let $RS = (X, \mathcal{R})$ be a range space and $B \subseteq X$. The empirical VC-dimension $EVC(RS, B)$ of RS on B is the maximum cardinality of a subset of B shattered by $\mathcal{R}$. The VC-dimension $VC(RS)$ of RS is defined as $VC(RS) = EVC(RS, X)$.*

Example 2. *Let $X = [0, 1]$ be the set of all the points in $[0, 1]$ and let $\mathcal{R}$ be the set of subsets $[a, b]$, with $0 \leq a \leq b \leq 1$, that is $[a, b] \subseteq [0, 1]$. Let us consider the set $Y = \{x, y, z\}$, containing 3 points $0 \leq x < y < z \leq 1$. It is not possible to find a range whose intersection with the set Y is $\{x, z\}$, since all the ranges $[a, b]$, with $0 \leq a \leq b \leq 1$, containing x and z, also contain y. Then, $VC(X, \mathcal{R})$ must be less than 3. Consider now the set $Y = \{x, y\}$, containing only 2 points $0 \leq x < y \leq 1$. It is easy to see that Y is shattered by $\mathcal{R}$, so $VC(X, \mathcal{R}) = 2$.*

The main application of VC-dimension in statistics and learning theory is to derive the sample size needed to approximately "learn" the ranges, as defined below.

Definition 6. *Let $RS = (X, \mathcal{R})$ be a range space. Given $\varepsilon \in (0, 1)$, a bag B of elements taken from X is an ε-bag of X if for all $r \in \mathcal{R}$, we have*

$$\left| \frac{|X \cap r|}{|X|} - \frac{|B \cap r|}{|B|} \right| \leq \varepsilon. \tag{10}$$

Theorem 1. *There is a constant $c > 0$ such that if $(X, \mathcal{R})$ is a range space of VC-dimension $\leq d$, and $\varepsilon, \delta \in (0, 1)$, then a bag B of m elements taken with independent random extractions with replacement from X, where*

$$m \geq \frac{c}{\varepsilon^2} \left(d + \ln \frac{1}{\delta} \right), \tag{11}$$

is an ε-bag of X with probability $\geq 1 - \delta$.

The universal constant c has been experimentally estimated to be at most 0.5 [24]. In the remaining of this work, we will use $c = 0.5$. Note that Theorem 1 holds also when d is an upper bound to the empirical VC-dimension $EVC(RS, B)$ of RS on B [25]. In that case, the bag B itself is an ε-bag of X.

2.3. Rademacher Complexity

The Rademacher complexity [3,23,26] is a tool to measure the complexity of a family of real-valued functions. Bounds based on the Rademacher complexity depend on the distribution of the dataset, differently from the ones based on VC-dimension that are distribution independent.

Let $\mathcal{D}$ be a dataset of n transactions $\mathcal{D} = \{t_1, \ldots, t_n\}$. For each $i \in \{1, \ldots, n\}$, let σ_i be an independent Rademacher random variable (r.v.) that takes value 1 or -1, each with probability $1/2$. Let $\mathcal{G}$ be a set of real-valued functions. The empirical Rademacher complexity $R_{\mathcal{D}}$ on $\mathcal{D}$ is defined as follows:

$$R_{\mathcal{D}} = \mathbb{E}_{\sigma} \left[\sup_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i g(t_i) \right], \tag{12}$$

where the expectation is taken w.r.t. the Rademacher r.v. σ_i's.

The Rademacher complexity is a measure of the expressiveness of the set $\mathcal{G}$. A specific combination of σ's represents a splitting of $\mathcal{D}$ into two random sub-samples $\mathcal{D}_1$ and $\mathcal{D}_{-1}$. For a function $g \in \mathcal{G}$, $\sum_{i=1}^{n} g(t_i)/n$ represents a good approximation of $\mathbb{E}[g]$ over $\mathcal{D}$ if n is sufficiently large. $\sum_{i=1}^{n} \sigma_i g(t_i)/n$ represents instead the difference between $\mathbb{E}[g]$ over the two random sub-samples $\mathcal{D}_1$ and $\mathcal{D}_{-1}$. By considering the expected value of the supremum of this difference over the set $\mathcal{G}$, we get the empirical Rademacher complexity. *Therefore, the intuition* is that if $R_{\mathcal{D}}$ is small, the dataset $\mathcal{D}$ is sufficiently large to ensure a good estimate of $\mathbb{E}[g]$ for every $g \in \mathcal{G}$. In this work, we study the Rademacher complexity of sequential patterns, which has not been explored before.

2.4. Maximum Deviation

Let $\mathcal{M}$ be a probability distribution over a domain set $\mathcal{Z}$. Let $\mathcal{F}$ be a set of functions that go from $\mathcal{Z}$ to $[-1,1]$. Given a function $f \in \mathcal{F}$, we define the expectation of f as:

$$\mathbb{E}(f) = \mathbb{E}_{z\sim\mathcal{M}}[f(z)], \tag{13}$$

and, given a sample Z of n observations $z_1, \ldots, z_n$ drawn from $\mathcal{M}$, the empirical average of f on Z as:

$$E(f,Z) = \frac{1}{n}\sum_{i=1}^{n} f(z_i). \tag{14}$$

The *maximum deviation* is defined as the largest difference between the expectation of a function f and its empirical average on sample Z as:

$$\sup_{f\in\mathcal{F}} |\mathbb{E}(f) - E(f,Z)|. \tag{15}$$

We now use the maximum deviation to capture quantities of interest for the two mining tasks we consider in this work.

In the frequent pattern mining scenario, we aim to find good estimates for $f_{\mathcal{D}}(p)$ for each pattern p. The frequency $f_{\mathcal{D}}(p)$ is the expectation of a Bernoulli random variable (r.v.) $X_{\mathcal{D}}(p,t)$ which is 1 if the pattern p appears in a transaction t drawn uniformly at random from $\mathcal{D}$:

$$\mathbb{E}_{t\sim\mathcal{D}}[X_{\mathcal{D}}(p,t)] = \Pr_{t\sim\mathcal{D}}(X_{\mathcal{D}}(p,t) = 1) = Supp_{\mathcal{D}}(p)/|\mathcal{D}| = f_{\mathcal{D}}(p). \tag{16}$$

Let $\mathcal{S}$ be a sample of transactions drawn uniformly and independently at random from $\mathcal{D}$. We define the frequency $f_{\mathcal{S}}(p)$ as the fraction of transactions of $\mathcal{S}$ where p appears. In this scenario, we have that the frequency $f_{\mathcal{D}}(p)$ of p on $\mathcal{D}$ and the frequency $f_{\mathcal{S}}(p)$ of p on the sample $\mathcal{S}$ represent, respectively, the expectation $\mathbb{E}(f_p)$ and the empirical average $E(f_p,\mathcal{S})$ of a function f_p associated with a pattern p. Thus, the maximum deviation is:

$$\sup_{p\in\mathbb{U}} |f_{\mathcal{D}}(p) - f_{\mathcal{S}}(p)|. \tag{17}$$

In the true frequent pattern mining scenario, we aim to find good estimates for $t_\pi(p)$ for each pattern p. Note that the true frequency $t_\pi(p)$ is the expectation of a Bernoulli r.v. which is 1 if the pattern p appears in a transaction drawn from π. Moreover, it is easy to prove that the observed frequency $f_{\mathcal{D}}(p)$ of a pattern p in a dataset $\mathcal{D}$ of transactions drawn from π is an unbiased estimator for $t_\pi(p)$, that is: $\mathbb{E}\left[f_{\mathcal{D}}(p)\right] = t_\pi(p)$.

Therefore, the true frequency $t_\pi(p)$ and the frequency $f_{\mathcal{D}}(p)$ observed on the dataset $\mathcal{D}$ represent, respectively, the expectation $\mathbb{E}(f_p)$ and the empirical average $E(f_p,\mathcal{D})$ of a function f_p associated with a pattern p. Thus, the maximum deviation is:

$$\sup_{p\in\mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)|. \tag{18}$$

In the next sections, we provide probabilistic upper bounds to the maximum deviation using the VC-dimension and Rademacher complexity which can therefore be used for frequent pattern mining and true frequent pattern mining scenarios.

3. VC-Dimension of Sequential Patterns

In this section, we apply the statistical learning theory concept of VC-dimension to sequential patterns. First, we define the range space associated with a sequential dataset. Then, we show a computable efficient upper bound on the VC-dimension and, finally, we present two applications of such upper bound. The first one is to compute the size of a sample that guarantees to obtain a good approximation for the problem of mining the frequent sequential patterns. The second one is to compute an upper bound on the maximum deviation to mine the true frequent sequential patterns.

Remember that a range space is a pair $(X, \mathcal{R})$ where X contains points and $\mathcal{R}$ contains ranges. For a sequential dataset, X is the dataset itself, while $\mathcal{R}$ contains the sequential transactions that are the support set for some sequential patterns.

Definition 7. *Let $\mathcal{D}$ be a sequential dataset consisting of sequential transactions and let $\mathcal{I}$ be its ground set. Let $\mathbb{U}$ be the set of all sequences built with itemsets containing item from $\mathcal{I}$. We define $RS = (X, \mathcal{R})$ to be a range space associated with $\mathcal{D}$ such that:*

- *$X = \mathcal{D}$ is the set of sequential transactions in the dataset;*
- *$\mathcal{R} = \{T_{\mathcal{D}}(p) : p \in \mathbb{U}\}$ is a family of sets of sequential transactions such that for each sequential pattern p, the set $T_{\mathcal{D}}(p) = \{\tau \in \mathcal{D} : p \sqsubseteq \tau\}$ is the support set of p on $\mathcal{D}$.*

The VC-dimension of this range space is the maximum size of a set of sequential transactions that can be shattered by the support sets of the sequential patterns.

Example 3. *Consider the following dataset $\mathcal{D} = \{\tau_1, \tau_2, \tau_3, \tau_4\}$ as an example:*

$$\begin{aligned}
\tau_1 &= \langle \{1\}, \{2,3\}, \{4,5,6\} \rangle \\
\tau_2 &= \langle \{1\}, \{3\}, \{4\} \rangle \\
\tau_3 &= \langle \{7\}, \{3,4\} \rangle \\
\tau_4 &= \langle \{4\}, \{5\} \rangle
\end{aligned}$$

The dataset above has 4 transactions. We now show that the VC-dimension of the range space RS associated with $\mathcal{D}$ is 2. Let us consider the set $A = \{\tau_2, \tau_3\}$. The power set 2^A of A is $2^A = \{\emptyset, \{\tau_2\}, \{\tau_3\}, \{\tau_2, \tau_3\}\}$. A is shatter by $\mathcal{R}$ since the projection $P_{\mathcal{R}}(A)$ of $\mathcal{R}$ in A is equal to 2^A (remember that $P_{\mathcal{R}}(A) = \{r \cap A \ : \ r \in \mathcal{R}\}$):

$$\begin{aligned}
\emptyset &= A \cap T_{\mathcal{D}}(\langle \{6\} \rangle), \\
\{\tau_2\} &= A \cap T_{\mathcal{D}}(\langle \{1\} \rangle), \\
\{\tau_3\} &= A \cap T_{\mathcal{D}}(\langle \{3,4\} \rangle), \\
A = \{\tau_2, \tau_3\} &= A \cap T_{\mathcal{D}}(\langle \{3\} \rangle).
\end{aligned}$$

Since $|A| = 2$ and A is shattered by $\mathcal{R}$, then the range space associated with $\mathcal{D}$ has VC-dimension ≥ 2. Analogously, the sets $\{\tau_1, \tau_3\}$,$\{\tau_1, \tau_4\}$,$\{\tau_2, \tau_4\}$ and $\{\tau_3, \tau_4\}$ are shattered by $\mathcal{R}$. The set $B = \{\tau_1, \tau_2\}$ is instead not shattered by $\mathcal{R}$: since $\tau_2 \sqsubseteq \tau_1$, there is not a sequential pattern p^* such that $B \cap T_{\mathcal{D}}(p^*) = \{\tau_2\}$. The sets $C = \{\tau_1, \tau_3, \tau_4\}$ and $E = \{\tau_2, \tau_3, \tau_4\}$ are not shattered by $\mathcal{R}$ either: there is not a sequential pattern p' such that $\{\tau_3, \tau_4\} = C \cap T_{\mathcal{D}}(p')$ or $\{\tau_3, \tau_4\} = E \cap T_{\mathcal{D}}(p')$. Thus, the VC-dimension of the range space associated with $\mathcal{D}$ is exactly 2.

The exact computation of the (empirical) VC-dimension of the range space associated with a dataset $\mathcal{D}$ is computationally expensive. The s-index, introduced by Servan-Schreiber et al. [8], provides an efficiently computable upper bound on the VC-dimension of sequential patterns. Such upper bound is based on the notion of *capacity* $c(p)$ of a sequence p. The capacity $c(p)$ of a sequence p is the number of distinct subsequences of p, that is, $c(p) = |\{z : z \sqsubseteq p\}|$. The exact capacity can be computed using the algorithm described in Reference [27], but it is computationally expensive and may be prohibitive for large datasets. Instead, Reference [8] proposed an algorithm to compute a more efficient upper bound $\tilde{c}(p) \geq c(p)$. Let us consider that a first simple bound is given by $2^{||p||} - 1$, that may be a loose upper bound of $c(p)$ because it is obtained by considering all the items contained in all the itemsets in p as distinct, that is, the capacity of the sequence p is $2^{||p||} - 1$ if and only if all the items contained in all the itemsets of the sequence p are different. The bound proposed by Reference [8] can be computed as follows. When p contains, among others, two itemsets A and B such that $A \subseteq B$, subsequences of the form $\langle C \rangle$ with $C \subseteq A$ are considered twice in $2^{||p||} - 1$, "generated" once from A and once from B. To avoid over-counting such $2^{|A|} - 1$ subsequences, Reference [8] proposes to consider only the ones "generated" from the longest itemset that can generate them. Then, the s-index is defined as follows.

Definition 8 ([8])**.** *Let $\mathcal{D}$ be a sequential dataset. The* s-index *of $\mathcal{D}$ is the maximum integer s such that $\mathcal{D}$ contains at least s different sequential transactions with upper bound to their capacities $\tilde{c}(p)$ at least $2^s - 1$, such that no one of them is a subset of another, that is the s sequential transactions form an anti-chain.*

The following result from Reference [8] shows that the s-index is an upper bound to the VC-dimension of the range space for sequential patterns in $\mathcal{D}$.

Theorem 2 (Lemma 3 [8])**.** *Let $\mathcal{D}$ be a sequential dataset with s-index s. Then, the range space $RS = (X, \mathcal{R})$ corresponding to $\mathcal{D}$ has VC-dimension $\leq s$.*

While an upper bound to the s-index can be computed in a streaming fashion, it still requires to check whether a transaction is a subset of the set of other transactions currently maintained in memory and that define the current value of the s-index. In addition, the computation of the upper bound $\tilde{c}(p)$ on the capacity of a sequence p requires to check whether the itemsets of p are subsets of each others. To avoid such expensive operations, we define an upper bound to the s-index, that we call *s-bound*, which does not require to check whether the transactions form an anti-chain.

Definition 9. *Let $\mathcal{D}$ be a sequential dataset. The* s-bound *of $\mathcal{D}$ is the maximum integer s such that $\mathcal{D}$ contains at least s different sequential transactions with item-length at least s.*

Algorithm 1 shows the pseudo-code to compute an upper bound to the s-bound in a streaming fashion. It uses an ordered set to maintain in memory the set of transactions that define the current value of the s-bound. The ordered set stores pairs composed by a transaction and its item-length, sorted by decreasing item-length. In addition, it uses a hash set to speed up the control on the equal transactions.

In practice, it is quite uncommon that the long sequences that define the value of the s-index are subsequences of other sequences, thus, removing the anti-chain constraint, the bound does not deteriorate. In addition, the usage of the naive algorithm to compute the upper bound on $c(p)$, that is $2^{||p||} - 1$, it is equivalent to consider the transactions that have item-length at least s to calculate the s-bound, making the computation much faster without worsening the bound on the VC-dimension in practice.

Algorithm 1: SBoundUpp($\mathcal{D}$): computation of an upper bound on the s-bound.

```
Data: Dataset $\mathcal{D}$.
Result: Upper bound $d$ on the $s$-bound of $\mathcal{D}$.
$\mathcal{H} \leftarrow$ empty HashSet of transactions;
$\mathcal{O} \leftarrow$ empty OrderedSet of pairs (transaction, itemLength) sorted by decreasing itemLength;
$d \leftarrow 0$;
foreach $\tau \in \mathcal{D}$ do
    if $\tau \notin \mathcal{H}$ then
        $\ell \leftarrow$ ComputeItemLength($\tau$);
        if $\ell > d$ then
            $\mathcal{H}$.add($\tau$);
            $\mathcal{O}$.add(($\tau, \ell$));
            $(\tau', \ell') \leftarrow \mathcal{O}$.last();
            if $\ell' > d$ then $d \leftarrow d + 1$;
            else
                $\mathcal{H}$.remove($\tau'$);
                $\mathcal{O}$.removeLast();
return $d$;
```

3.1. Compute the Sample Size for Frequent Sequential Pattern Mining

In this section, we show how to compute a sample size m for a random sample S of transactions taken from $\mathcal{D}$ such that the maximum deviation is bounded by $\varepsilon/2$, that is, $\sup_{p\in\mathbb{U}} |f_{\mathcal{D}}(p) - f_S(p)| \leq \varepsilon/2$, for a user-defined value ε, using the upper bound on the VC-dimension defined above. Such result underlies the sampling algorithm that will be introduced in Section 5. Algorithm 2 shows how to compute a sample size that guarantees that $\sup_{p\in\mathbb{U}} |f_{\mathcal{D}}(p) - f_S(p)| \leq \varepsilon/2$ with probability $\geq 1 - \delta$. This algorithm is used in the sampling algorithm (Section 5).

Theorem 3 (Proof in Appendix A)**.** *Let S be a random sample of m transactions taken with replacement from the sequential dataset $\mathcal{D}$ and $\varepsilon, \delta \in (0,1)$. Let d be the s-bound of $\mathcal{D}$. If*

$$m \geq \frac{2}{\varepsilon^2}\left(d + \ln\frac{1}{\delta}\right), \tag{19}$$

then $\sup_{p\in\mathbb{U}} |f_{\mathcal{D}}(p) - f_S(p)| \leq \varepsilon/2$ with probability at least $1 - \delta$.

Algorithm 2: ComputeSampleSize($\mathcal{D}, \varepsilon, \delta$): computation of the sample size such that $\sup_{p\in\mathbb{U}} |f_{\mathcal{D}}(p) - f_S(p)| \leq \varepsilon/2$ with probability $\geq 1 - \delta$.

```
Data: Dataset $\mathcal{D}$; $\varepsilon, \delta \in (0,1)$.
Result: The sample size $m$.
$d \leftarrow$ SBoundUpp($\mathcal{D}$);
$m \leftarrow 2/\varepsilon^2 \, (d + \ln(1/\delta))$;
return $m$;
```

3.2. Compute an Upper Bound to the Max Deviation for the True Frequent Sequential Patterns

In this section, we show how to compute an upper bound on the maximum deviation $\mu_{VC}/2$ for the true frequent sequential pattern mining problem, that is, $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq \mu_{VC}/2$, using the upper bound on the empirical VC-dimension. Such result underlies the strategy for mining the true frequent sequential patterns that will be introduced in Section 6.

We define a range space associated with the generative process π as a range space where the points $X = \mathbb{U}$ and the range set $\mathcal{R} = \{T(p) : p \in \mathbb{U}\}$. The s-bound of the dataset $\mathcal{D}$, as defined above, is an upper bound on the empirical VC-dimension of the range space associated with π computed on $\mathcal{D}$. Algorithm 3 shows how to compute an upper bound on the maximum deviation that is used in the true frequent sequential pattern mining algorithm (Section 6).

Theorem 4 (Proof in Appendix A)**.** *Let $\mathcal{D}$ be a finite bag of $|\mathcal{D}|$ i.i.d. samples from an unknown probability distribution π on $\mathbb{U}$ and $\delta \in (0,1)$. Let d be the s-bound of $\mathcal{D}$. If*

$$\mu_{VC} = \sqrt{\frac{2}{|\mathcal{D}|}\left(d + \ln\frac{1}{\delta}\right)}, \tag{20}$$

then $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq \mu_{VC}/2$ with probability at least $1-\delta$.

Algorithm 3: ComputeMaxDevVC($\mathcal{D}$, δ): computation of an upper bound on the max deviation for the true frequent sequential pattern mining problem.

```
Data: Dataset $\mathcal{D}$; $\delta \in (0,1)$.
Result: Upper bound to the max deviation $\mu_{VC}/2$.
$d \leftarrow$ SBoundUpp($\mathcal{D}$);
$\mu_{VC} \leftarrow \sqrt{2/|\mathcal{D}|\,(d + \ln(1/\delta))}$;
return $\mu_{VC}/2$;
```

4. Rademacher Complexity of Sequential Patterns

In this section we introduce the Rademacher complexity of sequential patterns. We propose a method for finding an efficiently computable upper bound to the empirical Rademacher complexity $R_{\mathcal{D}}$ of sequential patterns (similar to what has been done in Reference [5] for itemsets) and a method for approximating it. In the true frequent pattern mining scenario, these results will be useful for defining a quantity which is an upper bound to the maximum deviation $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)|$ with high probability.

The introduction of the Rademacher complexity of sequential patterns requires the definition of a set of real-valued functions. We define, for each pattern $p \in \mathbb{U}$, the indicator function $\phi_p : \mathbb{U} \rightarrow \{0,1\}$ as:

$$\phi_p(t) = \begin{cases} 1 & if\ p \sqsubseteq t \\ 0 & otherwise \end{cases}, \tag{21}$$

where t is a transaction. Given a transaction t of a dataset $\mathcal{D}$ with n transactions, $\phi_p(t)$ is 1 if p appears in t, otherwise it is 0. We define the set of real-valued functions as the family of these indicator functions. The frequency of p in $\mathcal{D}$ can be defined using the indicator function ϕ_p: $f_{\mathcal{D}}(p) = \sum_{t\in\mathcal{D}} \phi_p(t)/n$. The *(empirical) Rademacher complexity $R_{\mathcal{D}}$* on a given dataset $\mathcal{D}$ is defined as:

$$R_{\mathcal{D}} = \mathbb{E}_\sigma \left[\sup_{p\in\mathbb{U}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \phi_p(t_i)\right], \tag{22}$$

where the expectation is taken w.r.t. the Rademacher r.v. σ_i, that is, conditionally on the dataset $\mathcal{D}$. The connection between the Rademacher complexity of sequential patterns and the maximum deviation is given by the following theorem, which derives from standard results in statistical learning theory (Thm. 3.2 in Reference [3]).

Theorem 5. *With probability at least* $1-\delta$*:*

$$\sup_{p\in\mathbb{U}} |t_{\pi}(p) - f_{\mathcal{D}}(p)| \leq 2R_{\mathcal{D}} + \sqrt{\frac{2\ln(2/\delta)}{|\mathcal{D}|}} = \frac{\mu_R}{2}. \tag{23}$$

The naïve computation of the exact value of $R_{\mathcal{D}}$ is expensive since it requires to mine all patterns from $\mathcal{D}$ and to generate all possible 2^n combination values of the Rademacher variables for the computation of the expectation. In the next sections we present an efficiently computable upper bound on the Rademacher complexity of sequential patterns and a simple method that approximates it, which are useful to find, respectively, an upper bound and an approximation to $\mu_R/2$.

4.1. An Efficiently Computable Upper Bound to the Rademacher Complexity of Sequential Patterns

For any pattern $p \in \mathbb{U}$, let us define the following $|\mathcal{D}|$-dimensional vector

$$v_{\mathcal{D}}(p) = (\phi_p(t_1), \ldots, \phi_p(t_{|\mathcal{D}|})) \tag{24}$$

and let $V_{\mathcal{D}} = \{v_{\mathcal{D}}(p), p \in \mathbb{U}\}$, where $t_1, t_2, \ldots, t_{|\mathcal{D}|}$ are the $|\mathcal{D}|$ transactions of $\mathcal{D}$. Note that all the infinite sequences of the universe $\mathbb{U}$ which do not appear in $\mathcal{D}$ are associated with the vector $(0,\ldots,0)$ of $|\mathcal{D}|$ zeros. This implies the finiteness of the size of $V_{\mathcal{D}}$: $|V_{\mathcal{D}}| < \infty$. In addition, defining $|\mathbb{U}(\mathcal{D})|$ as the number of sequential patterns that appear in $\mathcal{D}$, we have that potentially $|V_{\mathcal{D}}| \ll |\mathbb{U}(\mathcal{D})|$, since there may be two or more patterns associated with the same vector $v_{\mathcal{D}} \in V_{\mathcal{D}}$ (i.e., these patterns appear exactly in the same transactions).

The following two theorems derive from known results of statistical learning theory (Thm. 3.3 of Reference [3]). Both theorems have been used for mining frequent itemsets [5], and can be applied for sequential pattern mining.

Theorem 6. *(Massart's Lemma)*

$$R_{\mathcal{D}} \leq \max_{p\in\mathbb{U}} ||v_{\mathcal{D}}(p)|| \frac{\sqrt{2\ln|V_{\mathcal{D}}|}}{|\mathcal{D}|} \tag{25}$$

where $||\cdot||$ *indicates the Euclidean norm.*

The following theorem is a stronger version of the previous one.

Theorem 7. *Let* $w : \mathbb{R}^+ \to \mathbb{R}^+$ *be the function*

$$w(s) = \frac{1}{s} \ln \sum_{v\in V_{\mathcal{D}}} \exp\left(\frac{s^2||v||^2}{2|\mathcal{D}|^2}\right), \tag{26}$$

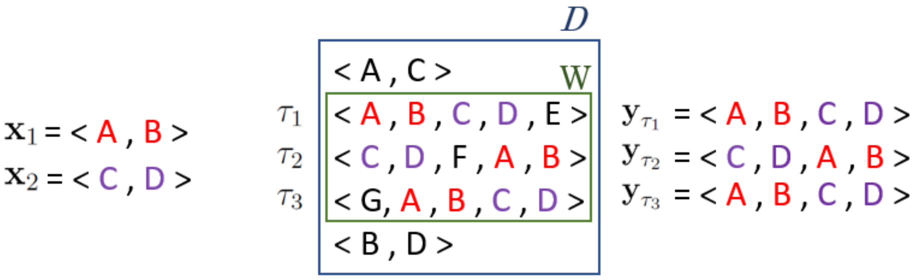
then

$$R_{\mathcal{D}} \leq \min_{s\in\mathbb{R}^+} w(s). \tag{27}$$

The upper bound on $R_{\mathcal{D}}$ of Theorem 7 is not directly applicable to sequential pattern mining since it requires to mine every pattern that appear in $\mathcal{D}$ in order to determine the entire set $V_{\mathcal{D}}$. However, the set $V_{\mathcal{D}}$ is related to the set of closed sequential patterns on $\mathcal{D}$. The following two results give us an upper bound to the size of $V_{\mathcal{D}}$ which depends on the number of closed sequential patterns of $\mathcal{D}$.

Lemma 1 (Proof in Appendix A)**.** *Consider a subset* W *of the dataset* $\mathcal{D}$*,* $W \subseteq \mathcal{D}$*. Let* $CS_W(\mathcal{D})$ *be the set of closed sequential patterns in* $\mathcal{D}$ *whose support set in* $\mathcal{D}$ *is* W*, that is,* $CS_W(\mathcal{D}) = \{p \in CS(\mathcal{D}) : T_{\mathcal{D}}(p) = W\}$*, with* $C = |CS_W(\mathcal{D})|$*. Then the number* C *of closed sequential patterns in* $\mathcal{D}$ *with* W *as support set satisfies:* $0 \leq C \leq |CS(\mathcal{D})|$*.*

A simple example where $C = 2$ is depicted in Figure 1. Note first of all that each super-sequence of x_1 but not of x_2 has support lower than the support of x_1, and each super-sequence of x_2 but not of x_1 has support lower than the support of x_2. Let $\mathbf{y}_\tau = \tau_{x_1,x_2}$ be the subsequence of transaction τ restricted to only the sequences x_1 and x_2, preserving the relative order of their itemsets. Then $y_{\tau_1} = y_{\tau_3} \neq y_{\tau_2}$ which implies $|T_W(y_{\tau_1})|$, $|T_W(y_{\tau_2})|$, and $|T_W(y_{\tau_3})|$ be lower than $|T_W(x_1)| = |T_W(x_2)| = |W|$. Therefore each super-sequence of both x_1 and x_2 has support lower than the support of x_1 (i.e. equal to the one of x_2). Thus, x_1 and x_2 are closed sequences in $\mathcal{D}$ with the same support set W.



$\mathcal{D}$
W
$\mathbf{x}_1 = <A, B>$
$\mathbf{x}_2 = <C, D>$
<A , C>
τ_1 <A , B , C , D , E>
τ_2 <C , D , F , A , B>
τ_3 <G, A , B , C , D>
<B , D>
$\mathbf{y}_{\tau_1} = <A, B, C, D>$
$\mathbf{y}_{\tau_2} = <C, D, A, B>$
$\mathbf{y}_{\tau_3} = <A, B, C, D>$

Figure 1. Graphical representation of the case $CS_W(\mathcal{D}) = 2$. Sequences x_1 and x_2 are closed sequences in $\mathcal{D}$ with the same support set W.

Note that the previous lemma represents a sequential patterns version of Lemma 3 of Reference [5] for itemsets, where the upper bound to the number of closed itemsets in $\mathcal{D}$ with W as support set is one (this holds by the nature of the itemsets where the notion of "ordering" is not defined). Lemma 1 is crucial for proving the following lemma which provides a bound on the size of the set $V_\mathcal{D}$ of binary vectors.

Lemma 2 (Proof in Appendix A)**.** *$V_\mathcal{D} = \{v_\mathcal{D}(p) : p \in CS(\mathcal{D})\} \cup \{(0, \ldots, 0)\}$ and $|V_\mathcal{D}| \leq |CS(\mathcal{D})| + 1$, that is, each vector of $V_\mathcal{D}$ different from $(0, \ldots, 0)$ is associated with at least one closed sequential pattern in $\mathcal{D}$.*

Combining a partitioning of $CS(\mathcal{D})$ with the previous lemma we can define a function $\tilde{w}$, an upper bound to the function w of Theorem 7, which is efficient to compute with a single scan of $\mathcal{D}$. Let $\mathcal{I}$ be the set of items that appear in the dataset $\mathcal{D}$ and $<_o$ be its increasing ordering by their support in $\mathcal{D}$ (ties broken arbitrarily). Given an item a, let $T_\mathcal{D}(\langle\{a\}\rangle)$ be its support set on $\mathcal{D}$. Let $<_a$ denote the increasing ordering of the transactions $T_\mathcal{D}(\langle\{a\}\rangle)$ by the number of items contained that come after a w.r.t. the ordering $<_o$ (ties broken arbitrarily). Let $CS(\mathcal{D}) = C_1 \cup C_{2+}$, where $C_1 = \{p \in CS(\mathcal{D}) : ||p|| = 1\}$ and $C_{2+} = \{p \in CS(\mathcal{D}) : ||p|| \geq 2\}$. Let us focus on partitioning C_{2+}. Let $p \in C_{2+}$ and let a be the item in p which comes before any other item in p w.r.t. the order $<_o$. Let τ be the transaction containing p which comes before any other transaction containing p w.r.t. the order $<_a$. We assign p to the set $C_{a,\tau}$. Remember that an item can appear multiple times in a sequence. Given a transaction $\tau \in T_\mathcal{D}(\langle\{a\}\rangle)$, $k_{a,\tau}$ is the number of items in τ (counted with their multiplicity) equal to a or that come after a in $<_o$. Let $m_{a,\tau}$ be the multiplicity of a in τ. For each $k, m \geq 1$, $m \leq k$, let $g_{a,k,m}$ be the number of transactions in $T_\mathcal{D}(\langle\{a\}\rangle)$ that contain exactly k items (counted with their multiplicity) equal to a or located after a in the ordering $<_o$, with exactly m repetitions of a. Let $\chi_a = max\{k : g_{a,k,m} > 0\}$. The following lemma gives us an upper bound to the size of $C_{a,\tau}$.

Lemma 3 (Proof in Appendix A)**.** *We have*

$$|C_{a,\tau}| \leq 2^{k_{a,\tau} - m_{a,\tau}}(2^{m_{a,\tau}} - 1). \tag{28}$$

Combining the following partitioning of $CS(\mathcal{D})$ as

$$CS(\mathcal{D}) = C_1 \cup C_{2+} = C_1 \cup \left(\bigcup_{a \in \mathcal{I}} \bigcup_{\tau \in T_{\mathcal{D}}(\langle \{a\} \rangle)} C_{a,\tau} \right) \tag{29}$$

with the previous lemma, we obtain

$$|CS(\mathcal{D})| \leq |\mathcal{I}| + \sum_{a \in \mathcal{I}} \sum_{\tau \in T_{\mathcal{D}}(\langle \{a\} \rangle)} 2^{k_{a,\tau} - m_{a,\tau}} (2^{m_{a,\tau}} - 1). \tag{30}$$

Now we are ready to define the function $\tilde{w}$, which can be used to obtain an efficiently computable upper bound to $R_{\mathcal{D}}$. The following lemma represents the analogous of Lemma 5 of Reference [5], adjusted for sequential patterns. Let $\overline{\eta}$ be the average item-length of the transactions of $\mathcal{D}$, that is, $\overline{\eta} = \sum_{t \in \mathcal{D}} ||t||/n$. Let $\hat{\eta}$ be the maximum item-length of the transactions of $\mathcal{D}$, that is, $\hat{\eta} = \max_{t \in \mathcal{D}} ||t||$. Let η be an item-length threshold, with $\overline{\eta} < \eta \leq \hat{\eta}$. Let $\mathcal{D}(\eta)$ be the bag of transactions of $\mathcal{D}$ with item-length greater than η. Let $V_{\mathcal{D}(\eta)}$ be the set of the $2^{|\mathcal{D}(\eta)|} - 1$ binary vectors associated with all possible non-empty sub-bags of $\mathcal{D}(\eta)$.

Lemma 4 (Proof in Appendix A)**.** *Given an item a in $\mathcal{I}$, we define the following quantity:*

$$q(a,\eta) = 1 + \sum_{k=1}^{\chi_a} \sum_{m=1}^{k} \sum_{j=1}^{g_{a,k,m}} \left(\mathbb{1}(k \leq \eta) 2^{k-m} (2^m - 1) + \mathbb{1}(k > \eta) \sum_{i=1}^{\eta - 1} \binom{k-1}{i} \right). \tag{31}$$

Let $\tilde{w} : \mathbb{R}^+ \to \mathbb{R}^+$ be the function

$$\tilde{w}(s,\eta) = \frac{1}{s} \ln \sum_{a \in \mathcal{I}} \left(q(a,\eta) e^{\frac{s^2 f_{\mathcal{D}}(\langle \{a\} \rangle)}{2|\mathcal{D}|}} + |V_{\mathcal{D}(\eta)}| e^{\frac{s^2 |\mathcal{D}(\eta)|}{2|\mathcal{D}|^2}} + 1 \right). \tag{32}$$

Then,

$$R_{\mathcal{D}} \leq \min_{s \in \mathbb{R}^+, \overline{\eta} < \eta \leq \hat{\eta}} \tilde{w}(s,\eta). \tag{33}$$

For a given value of η, the function $\tilde{w}$ can be compute with a single scan of the dataset, since it requires to know $g_{a,k,m}$ for each $a \in \mathcal{I}$ and for each k, m, $1 \leq k \leq \chi_a$, $1 \leq m \leq k$. The values $\overline{\eta}$, $\hat{\eta}$, and the support of each item and consequently the ordering $<_o$ are obtained during the dataset creation. Thus, it is sufficient to look at each transaction τ, sorting the items $\mathcal{I}_\tau$ that appear in τ according to $<_o$, and, for each item of $\mathcal{I}_\tau$, keep track of its multiplicity $m_{a,\tau}$, compute $k_{a,\tau}$ and increase by one $g_{a,k_{a,\tau},m_{a,\tau}}$. Finally, since $\tilde{w}$ is convex and has first and second derivatives w.r.t. s everywhere in $\mathbb{R}^+$, its global minimum can be computed using a non-linear optimization solver. This procedure has to be repeated for each possible value of η in $(\overline{\eta}, \hat{\eta}]$.

However, one could choose a particular schedule of values of η to be tested, instead of taking into account each possible value, achieving a value of the function $\tilde{w}$ near to its minimum. A possible choice is to look at the restricted interval $[\overline{\eta} + \beta_1, \min(\beta_2, \hat{\eta})]$, given two positive values for β_1 and β_2, instead of investigating the whole interval $(\overline{\eta}, \hat{\eta}]$. This choice is motivated by the fact that in Lemma 4 the value of η gives us an idea of which term of the summation is dominant (the one based on closed sequential patterns or the one based on binary vectors). If η is close to $\overline{\eta}$ then the number of binary vectors we count could be high, the dominant term is the one based on the set of binary vectors, and we expect the upper bound to be high. Instead, if η is close to $\hat{\eta}$ then the upper bound to the number of closed sequential patterns we count could be high, and the set of binary vectors we take into account is small. In this case, the dominant term is the one based on the closed sequential patterns, and the value of the upper bound could be high (since we count many sequential patterns with item-length greater than η that instead would be associated with a small number of binary vectors). Thus, the best value of η will be the one that is larger than $\overline{\eta}$ and smaller than $\hat{\eta}$, enough to count not too many closed sequential patterns and binary vectors.

Finally, we define *ComputeMaxDevRadeBound* as the procedure for computing an upper bound to $\mu_R/2$ where, once the upper bound $R^b_{\mathcal{D}}$ to the Rademacher complexity $R_{\mathcal{D}}$ is computed using Algorithm 4, the upper bound $\mu^b_R/2$ to $\mu_R/2$ is obtained by

$$\frac{\mu^b_R}{2} = 2R^b_{\mathcal{D}} + \sqrt{\frac{2\ln(2/\delta)}{|\mathcal{D}|}}. \tag{34}$$

The pseudo-code of the algorithm for computing the upper bound to $R_{\mathcal{D}}$ follows.

Algorithm 4: RadeBound($\mathcal{D}$): algorithm for bounding the empirical Rademacher complexity of sequential patterns

```
Data: : a sequential dataset $\mathcal{D}$ built on alphabet $\mathcal{I}$
Result: upper bound to $R_{\mathcal{D}}$
$g_{a,k,m} \leftarrow 0, \forall a \in \mathcal{I},\ k,m \in \mathbb{N},\ m \le k$;
$\chi_a \leftarrow 0, \forall a \in \mathcal{I}$;
/* $\overline{\eta}$, $\hat{\eta}$, and the support of the items are computed during the scan of $\mathcal{D}$ */
for $\tau \in \mathcal{D}$ do
    for $a \in \tau$ do
        $k_{a,\tau} \leftarrow$ number of items in $\tau$ (counted with their multiplicity) equal to $a$ or that come
          after $a$ in $<_o$;
        $m_{a,\tau} \leftarrow$ number of repetitions of $a$ in $\tau$;
        $g_{a,k_{a,\tau},m_{a,\tau}} += 1$;
        $\chi_a \leftarrow \max(\chi_a, k_{a,\tau})$;
return $\min_{s\in\mathbb{R}^+,\overline{\eta}<\eta\le\hat{\eta}} \tilde{w}(s,\eta)$;
```

4.2. Approximating the Rademacher Complexity of Sequential Patterns

The previous section presents an efficiently computable upper bound to the Rademacher of sequential patterns, which does not require any extraction of frequent sequences from a given dataset. Here we present a simple method that gives us an approximation of the Rademacher complexity of sequential patterns, which provides a tighter bound to the maximum deviation compared to the ones previously presented.

In the definition of the Rademacher complexity, a given combination $\overline{\sigma}$ of the Rademacher r.v. σ splits the dataset $\mathcal{D}$ of n transactions in two sub-samples $\mathcal{D}_1(\overline{\sigma})$ and $\mathcal{D}_{-1}(\overline{\sigma})$: each transaction associated with 1 and -1 goes respectively into $\mathcal{D}_1(\overline{\sigma})$ and $\mathcal{D}_{-1}(\overline{\sigma})$. For a given sequential pattern $p \in \mathbb{U}$, let $Supp_{\mathcal{D}_1(\overline{\sigma})}(p)$ and $Supp_{\mathcal{D}_{-1}(\overline{\sigma})}(p)$ be respectively the number of transactions of $\mathcal{D}_1(\overline{\sigma})$ and $\mathcal{D}_{-1}(\overline{\sigma})$ in which p appears. Thus, the Rademacher complexity can be rewritten as follows:

$$R_{\mathcal{D}} = \mathbb{E}_\sigma\left[\sup_{p\in\mathbb{U}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \phi_p(t_i)\right] = \mathbb{E}_\sigma\left[\sup_{p\in\mathbb{U}} \frac{Supp_{\mathcal{D}_1(\sigma)}(p) - Supp_{\mathcal{D}_{-1}(\sigma)}(p)}{n}\right]. \tag{35}$$

In our approximation method we generate a single combination $\overline{\sigma}$ of the Rademacher r.v. σ, instead of generating every possible combination and then taking the expectation. Given $\overline{\sigma}$, the approximation $\tilde{R}_{\mathcal{D}}(\overline{\sigma})$ of $R_{\mathcal{D}}$ is

$$\tilde{R}_{\mathcal{D}}(\overline{\sigma}) = \sup_{p\in\mathbb{U}} \frac{Supp_{\mathcal{D}_1(\overline{\sigma})}(p) - Supp_{\mathcal{D}_{-1}(\overline{\sigma})}(p)}{n}. \tag{36}$$

The first step of the procedure is to mine frequent sequential patterns from $\mathcal{D}_1(\overline{\sigma})$ and $\mathcal{D}_{-1}(\overline{\sigma})$, given a frequency threshold κ. Let $FSP(\mathcal{D}_1(\overline{\sigma}), \kappa)$ and $FSP(\mathcal{D}_{-1}(\overline{\sigma}), \kappa)$ be the sets of sequential patterns with support greater or equal than κ in $\mathcal{D}_1(\overline{\sigma})$ and $\mathcal{D}_{-1}(\overline{\sigma})$, respectively. Let us define the following quantities:

$$\gamma(p) = Supp_{\mathcal{D}_1(\overline{\sigma})}(p) - Supp_{\mathcal{D}_{-1}(\overline{\sigma})}(p), \tag{37}$$

$$\gamma_1 = \sup\{\gamma(p) : p \in FSP(\mathcal{D}_1(\overline{\sigma}), \kappa) \cap FSP(\mathcal{D}_{-1}(\overline{\sigma}), \kappa)\}, \tag{38}$$

and

$$\gamma_2 = \sup\{\gamma(p) : p \in FSP(\mathcal{D}_1(\overline{\sigma}), \kappa) \setminus FSP(\mathcal{D}_{-1}(\overline{\sigma}), \kappa)\}. \tag{39}$$

If $\max(\gamma_1, \gamma_2)/n \geq \kappa$ then $\tilde{R}_{\mathcal{D}}(\overline{\sigma}) = \max(\gamma_1, \gamma_2)/n$, since each pattern p that is not frequent in both sub-samples has $\gamma(p)/n$ lower than κ. Instead, if $\max(\gamma_1, \gamma_2)/n < \kappa$ the entire procedure is repeated with $\kappa = \max(\gamma_1, \gamma_2)/n$. Note that, since the Rademacher complexity is a non-negative quantity, it is not necessary to look at patterns in $FSP(\mathcal{D}_{-1}(\overline{\sigma}), \kappa) \setminus FSP(\mathcal{D}_1(\overline{\sigma}), \kappa)$ since their $\gamma(p)$'s values are negative. The pseudo-code of the method for finding an approximation of $R_{\mathcal{D}}$ is presented in Algorithm 5. The extraction of frequent sequences from the two sub-samples can be done using one of the many algorithms for mining frequent sequential patterns.

Algorithm 5: RadeApprox($\mathcal{D}, \kappa$): algorithm for approximating the Rademacher complexity of sequential patterns.

```
Data: : dataset $\mathcal{D}$; $\kappa \in (0,1]$
Result: approximation to $R_{\mathcal{D}}$
$\overline{\sigma} \leftarrow$ combination of $\sigma$;
split $\mathcal{D}$ into $\mathcal{D}_1(\overline{\sigma})$ and $\mathcal{D}_{-1}(\overline{\sigma})$;
$found \leftarrow false$;
$\gamma \leftarrow 0$;
while $!found$ do
    compute $FSP(\mathcal{D}_1(\overline{\sigma}), \kappa)$;
    compute $FSP(\mathcal{D}_{-1}(\overline{\sigma}), \kappa)$;
    if $|FSP(\mathcal{D}_1(\overline{\sigma}), \kappa)| + |FSP(\mathcal{D}_{-1}(\overline{\sigma}), \kappa)| = 0$ then
        $\kappa \leftarrow \kappa/2$;
        continue;
    compute $\gamma_1$ and $\gamma_2$;
    $\gamma \leftarrow \max(\gamma_1, \gamma_2)/|\mathcal{D}|$;
    if $\gamma \geq \kappa$ then $found \leftarrow true$;
    else $\kappa \leftarrow \gamma$;
return $\gamma$;
```

Finally, we define *ComputeMaxDevRadeApprox* as the procedure for computing an approximation of $\mu_R/2$ where, once the approximation $R^a_{\mathcal{D}}$ of the Rademacher complexity $R_{\mathcal{D}}$ is computed using Algorithm 5, the approximation $\mu^a_R/2$ of $\mu_R/2$ is obtained by:

$$\frac{\mu^a_R}{2} = 2R^a_{\mathcal{D}} + \sqrt{\frac{2\ln(2/\delta)}{|\mathcal{D}|}}. \tag{40}$$

5. Sampling-Based Algorithm for Frequent Sequential Pattern Mining

We now present a sampling algorithm for frequent sequential pattern mining. The aim of this algorithm is to reduce the amount of data to consider to mine the frequent sequential patterns, in order to speed up the extraction of the sequential patterns and to reduce the amount of memory required. We define a *random sample* as a bag of m transactions taken uniformly and independently at random, with replacement, from $\mathcal{D}$. Obtaining the exact set $FSP(\mathcal{D},\theta)$ from a random sample is not possible, thus we focus on obtaining an ε-approximation with probability at least $1-\delta$, where $\delta \in (0,1)$ is a *confidence* parameter, whose value, with ε, is provided in input by the user. Intuitively, if a random sample is sufficiently large, then the set of frequent sequential patterns extracted from the random sample well approximates the set $FSP(\mathcal{D},\theta)$. The challenge is to find the number of transactions that are necessary to obtain the desired ε-approximation. To compute such sample size, our approach uses the VC-dimension of sequential patterns (see Section 3.1).

Theorem 8. *Given $\varepsilon, \delta \in (0,1)$, let S be a random sample of size m sequential transactions taken independently at random with replacement from the dataset $\mathcal{D}$ such that $\sup_{p \in \mathbb{U}} |f_{\mathcal{D}}(p) - f_S(p)| \leq \varepsilon/2$ with probability at least $1-\delta$. Then, given $\theta \in (0,1]$, the set $FSP(S,\theta-\varepsilon/2)$ is an ε-approximation to $FSP(\mathcal{D},\theta)$ with probability at least $1-\delta$.*

Proof. Suppose that $\sup_{p \in \mathbb{U}} |f_{\mathcal{D}}(p) - f_S(p)| \leq \varepsilon/2$. In such a scenario, we have that for all sequential patterns $p \in \mathcal{D}$, it results $f_S(p) \in [f_{\mathcal{D}}(p) - \varepsilon/2, f_{\mathcal{D}}(p) + \varepsilon/2]$. This also holds for the sequential patterns in $\mathcal{C} = FSP(S,\theta-\varepsilon/2)$. Therefore, the set $\mathcal{C}$ satisfies Property 3 from Definition 1. It also means that for all $p \in FSP(\mathcal{D},\theta)$, $f_S(p) \geq \theta - \varepsilon/2$, so such $p \in \mathcal{C}$ and $\mathcal{C}$ also satisfies Property 1. Now, let p^* be a sequential pattern such that $f_{\mathcal{D}}(p^*) < \theta - \varepsilon$. Then, $f_S(p^*) < \theta - \varepsilon/2$, that is $p^* \notin \mathcal{C}$, which allows us to conclude that $\mathcal{C}$ also has Property 2 from Definition 1. Since we know that $\sup_{p \in \mathbb{U}} |f_{\mathcal{D}}(p) - f_S(p)| \leq \varepsilon/2$ with probability at least $1-\delta$, then the set $\mathcal{C}$ is an ε-approximation to $FSP(\mathcal{D},\theta)$ with probability at least $1-\delta$, which concludes the proof. □

Theorem 8 provides a simple sampling-based algorithm to obtain an ε-approximation to $FSP(\mathcal{D},\theta)$ with probability $\geq 1-\delta$: take a random sample of m transactions from $\mathcal{D}$ such that the maximum deviation is bounded by $\varepsilon/2$, that is, $\sup_{p \in \mathbb{U}} |f_{\mathcal{D}}(p) - f_S(p)| \leq \varepsilon/2$; report in output the set $FSP(S,\theta-\varepsilon/2)$. As illustrated in Section 3.1, such sample size can be computed using an efficient upper bound on the VC-dimension, given in input the desired upper bound on the maximum deviation $\varepsilon/2$ (see Algorithm 2). Note that such sample size can not be computed with the Rademacher complexity, since the sample size appears in both terms of the right-hand side of Equation (23). Thus, it is not possible to fix the value of the bound on the maximum deviation to compute the sample size that provides such guarantees. Algorithm 6 shows the pseudo-code of the sampling algorithm.

We now provide the respective theorem to find a FPF ε-approximation.

Theorem 9. *Given $\varepsilon, \delta \in (0,1)$, let S be a random sample of size m sequential transactions taken independently at random with replacement from the dataset $\mathcal{D}$ such that $\sup_{p \in \mathbb{U}} |f_{\mathcal{D}}(p) - f_S(p)| \leq \varepsilon/2$ with probability $\geq 1-\delta$. Then, given $\theta \in (0,1]$, the set $FSP(S,\theta+\varepsilon/2)$ is a FPF ε-approximation to $FSP(\mathcal{D},\theta)$ with probability $\geq 1-\delta$.*

Proof. Suppose that $\sup_{p \in \mathbb{U}} |f_{\mathcal{D}}(p) - f_S(p)| \leq \varepsilon/2$. In such a scenario, we have that for all sequential patterns $p \in \mathcal{D}$, it results $f_S(p) \in [f_{\mathcal{D}}(p) - \varepsilon/2, f_{\mathcal{D}}(p) + \varepsilon/2]$. This also holds for the sequential patterns in $\mathcal{F} = FSP(S,\theta+\varepsilon/2)$. Therefore, the set $\mathcal{F}$ satisfies Property 3 from Definition 2. It also means that for all $p^* \notin FSP(\mathcal{D},\theta)$, $f_S(p^*) < \theta + \varepsilon/2$, so such $p^* \notin \mathcal{F}$ and $\mathcal{F}$ also satisfies Property 1. Now, let p' be a sequential pattern such that $f_{\mathcal{D}}(p') \geq \theta + \varepsilon$. Then, $f_S(p') \geq \theta + \varepsilon/2$, that is $p' \in \mathcal{F}$, which allows us to conclude that $\mathcal{F}$ also has Property 2 from Definition 2. Since we know that $\sup_{p \in \mathbb{U}} |f_{\mathcal{D}}(p) - f_S(p)| \leq \varepsilon/2$ with probability at least $1-\delta$, then the set $\mathcal{F}$ is a FPF ε-approximation to $FSP(\mathcal{D},\theta)$ with probability at least $1-\delta$, which concludes the proof. □

Algorithm 6: Sampling-Based Algorithm for Frequent Sequential Pattern Mining.

```
Data: Dataset $\mathcal{D}$; $\varepsilon, \delta \in (0,1)$; $\theta \in (0,1]$.
Result: Set $\mathcal{C}$ that is an $\varepsilon$-approximation (resp. a FPF $\varepsilon$-approximation) to $FSP(\mathcal{D}, \theta)$ with
        probability $\geq 1 - \delta$.
$m \leftarrow$ ComputeSampleSize($\mathcal{D}, \varepsilon, \delta$);
$S \leftarrow$ sample of $m$ transactions taken independently at random with replacement from $\mathcal{D}$;
$\mathcal{C} \leftarrow FSP(S, \theta - \varepsilon/2)$;    /* resp. $\theta + \varepsilon/2$ to obtain a FPF $\varepsilon$-approximation */
return $\mathcal{C}$;
```

As explained above, the sample size m can be computed with Algorithm 2 that uses an efficient upper bound on the VC-dimension of sequential patterns. Then, the sample is generated taking m transactions uniformly and independently at random, with replacement, from $\mathcal{D}$. Finally, the mining of the sample S can be performed with any efficient algorithm for the exact mining of frequent sequential patterns. Figure 2 depicts a block diagram representing the relations between the algorithms presented in this work.

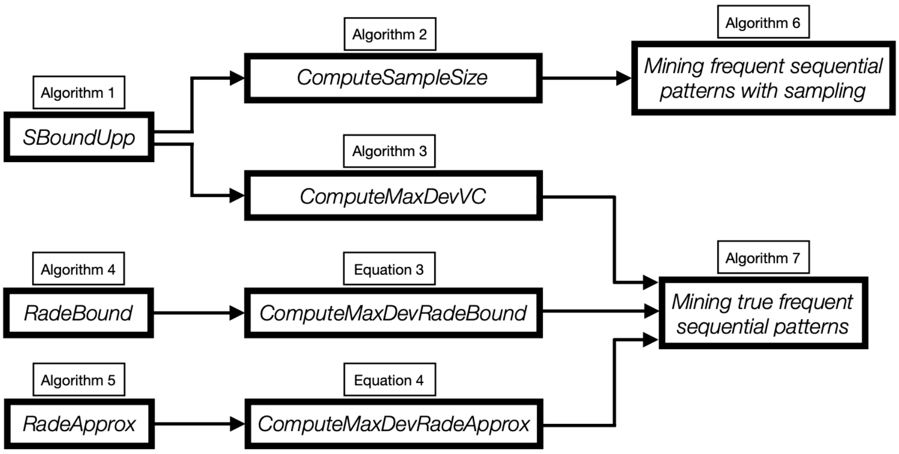


Figure 2. Block diagram representing the relations between our algorithms.

6. Algorithms for True Frequent Sequential Pattern Mining

In this section, we describe our approach to find rigorous approximations to the TFSPs. In particular, given a dataset $\mathcal{D}$, that is a finite bag of $|\mathcal{D}|$ i.i.d. samples from an unknown probability distribution π on $\mathbb{U}$, a minimum frequency threshold θ and a confidence parameter δ, we aim to find rigorous approximations of the TFSPs w.r.t. θ, defined in Definitions 3 and 4, with probability at least $1 - \delta$.

The intuition behind our approach is the following. If we know an upper bound $\mu/2$ on the maximum deviation, that is $\sup_{p \in \mathbb{U}} |t_\pi(p) - f_\mathcal{D}(p)| \leq \mu/2$, we can identify a frequency threshold $\hat{\theta}$ (resp. $\tilde{\theta}$) such that the set $FSP(\mathcal{D}, \hat{\theta})$ is a FPF μ-approximation (resp. $FSP(\mathcal{D}, \tilde{\theta})$ is a μ-approximation) of $TFSP(\pi, \theta)$. The upper bound on the maximum deviation can be computed, as illustrated in the previous sections, with the empirical VC-dimension and with the Rademacher complexity.

We now describe how to identify the threshold $\hat{\theta}$ that allows to obtain a FPF μ-approximation. Suppose that $\sup_{p \in \mathbb{U}} |t_\pi(p) - f_\mathcal{D}(p)| \leq \mu/2$. In such a scenario, we have that every sequential pattern $p^* \notin TFSP(\pi, \theta)$, and so that has $t_\pi(p^*) < \theta$, has a frequency $f_\mathcal{D}(p^*) < \theta + \mu/2 = \hat{\theta}$. Hence, the only sequential patterns that can have frequency in $\mathcal{D}$ greater or equal to $\hat{\theta} = \theta + \mu/2$, are those with true frequency at least θ. The intuition is that if we find a μ such that $\sup_{p \in \mathbb{U}} |t_\pi(p) - f_\mathcal{D}(p)| \leq \mu/2$, we know that all the sequences $p \in \mathbb{U}$, that are not true frequent w.r.t θ, can not be in $FSP(\mathcal{D}, \hat{\theta})$. The following theorem formalizes the strategy to obtain a FPF μ-approximation. Algorithm 7 shows the pseudo-code to mine the true frequent sequential patterns.

Theorem 10 shows how to compute a corrected threshold $\hat{\theta}$ such that the set $FSP(D, \hat{\theta})$ is a FPF μ-approximation of $TFSP(\pi, \theta)$, that is, $FSP(D, \hat{\theta})$ only contains sequential patterns that are in $TFSP(\pi, \theta)$. It guarantees that with high probability the set $FSP(\mathcal{D}, \hat{\theta})$ does not contain *false positives* but it has not guarantees on the number of *false negatives*, that is, sequential patterns that are in $TFSP(\pi, \theta)$ but not in $FSP(\mathcal{D}, \hat{\theta})$. On the other hand, we might be interested in finding all the true frequent sequential patterns in $TFSP(\pi, \theta)$. The following result shows how to identify a threshold $\tilde{\theta}$ such that the set $FSP(\mathcal{D}, \tilde{\theta})$ contains all the true frequent sequential patterns in $TFSP(\pi, \theta)$ with high probability, that is, $FSP(\mathcal{D}, \tilde{\theta})$ is a μ-approximation of $TFSP(\pi, \theta)$. Note that while Theorem 11 provides guarantees on false negatives, it does not provide guarantees on the number of false positives in $FSP(\mathcal{D}, \tilde{\theta})$.

Algorithm 7 shows the pseudo-code of the two strategies to mine the true frequent sequential patterns. To compute an upper bound on the maximum deviation, it is possible to use Algorithm 3 based on the empirical VC-dimension or the two procedures *ComputeMaxDevRadeBound* (Equation (34)) and *ComputeMaxDevRadeApprox* (Equation (40)) based on the Rademacher complexity. The mining of $\mathcal{D}$ can be performed with any efficient algorithm for the exact mining of frequent sequential patterns. Figure 2 shows the relations between the algorithms we presented for mining true frequent sequential patterns.

Theorem 10. *Given* $\delta \in (0,1)$*, such that* $\sup_{p \in \mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq \mu/2$ *with probability at least* $1 - \delta$*, and given* $\theta \in (0,1]$*, the set* $FSP(\mathcal{D}, \hat{\theta})$*, with* $\hat{\theta} = \theta + \mu/2$*, is a FPF* μ*-approximation of the set* $TFSP(\pi, \theta)$ *with probability at least* $1 - \delta$*.*

Proof. Suppose that $\sup_{p \in \mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq \mu/2$. Thus, we have that for all the sequential patterns $p \in \mathbb{U}$, it results $f_{\mathcal{D}}(p) \in [t_\pi(p) - \mu/2, t_\pi(p) + \mu/2]$. This also holds for the sequential patterns in $\mathcal{G} = FSP(\mathcal{D}, \hat{\theta})$. Therefore, the set $\mathcal{G}$ satisfies Property 3 of Definition 4. Let p^* be a sequential pattern such that $t_\pi(p^*) < \theta$, that is, it is not a true frequent sequential pattern w.r.t. θ. Then, $f_{\mathcal{D}}(p^*) < \theta + \mu/2 = \hat{\theta}$, that is, $p^* \notin \mathcal{G}$, which allows us to conclude that $\mathcal{G}$ also has Property 1 from Definition 4. Now, let p' be a sequential pattern such that $t_\pi(p') \geq \theta + \mu$. Then, $f_{\mathcal{D}}(p') \geq \theta + \mu/2$, that is $p' \in \mathcal{G}$, which allows us to conclude that $\mathcal{G}$ also has Property 2 from Definition 4. Since we know that $\sup_{p \in \mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq \mu/2$ with probability at least $1 - \delta$, then the set $\mathcal{G}$ is a FPF μ-approximation of $TFSP(\pi, \theta)$ with probability at least $1 - \delta$, which concludes the proof. □

Theorem 11. *Given* $\delta \in (0,1)$*, such that* $\sup_{p \in \mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq \mu/2$ *with probability at least* $1 - \delta$*, and given* $\theta \in (0,1]$*, the set* $FSP(\mathcal{D}, \tilde{\theta})$*, with* $\tilde{\theta} = \theta - \mu/2$*, is a* μ*-approximation of the set* $TFSP(\pi, \theta)$ *with probability at least* $1 - \delta$*.*

Proof. Suppose that $\sup_{p \in \mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq \mu/2$. Thus, we have that for all the sequential patterns $p \in \mathbb{U}$, it results $f_{\mathcal{D}}(p) \in [t_\pi(p) - \mu/2, t_\pi(p) + \mu/2]$. This also holds for the sequential patterns in $\mathcal{E} = FSP(\mathcal{D}, \tilde{\theta})$. Therefore, the set $\mathcal{E}$ satisfies Property 3 of Definition 3. It also means that for all $p \in TFSP(\pi, \theta)$, $f_{\mathcal{D}}(p) \geq \theta - \mu/2 = \tilde{\theta}$, that is, $p \in \mathcal{E}$, which allows us to conclude that $\mathcal{E}$ also has Property 1 from Definition 3. Now, let p^* be a sequential pattern such that $t_\pi(p^*) < \theta - \mu$. Then, $f_{\mathcal{D}}(p^*) < \theta - \mu/2$, that is $p^* \notin \mathcal{E}$, which allows us to conclude that $\mathcal{E}$ also has Property 2 from Definition 3. Since we know that $\sup_{p \in \mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq \mu/2$ with probability at least $1 - \delta$, then the set $\mathcal{E}$ is a μ-approximation of $TFSP(\pi, \theta)$ with probability at least $1 - \delta$, which concludes the proof. □

```
Algorithm 7: Mining the True Frequent Sequential Patterns.
    Data: Dataset $\mathcal{D}$;  $\delta \in (0,1)$;  $\theta \in (0,1]$
    Result: Set $\mathcal{G}$ that is a FPF $\mu$-approximation (resp. $\mu$-approximation) to $TFSP(\pi,\theta)$ with
            probability $\geq 1-\delta$.
    $\mu/2 \leftarrow$ ComputeMaxDeviationBound($\mathcal{D},\delta$);
    $\mathcal{G} \leftarrow FSP(\mathcal{D},\theta+\mu/2)$;    /* resp. $\theta-\mu/2$ to obtain a $\mu$-approximation */
    return $\mathcal{P}$;
```

7. Experimental Evaluation

In this section, we report the results of our experimental evaluation on multiple datasets to assess the performance of the algorithms we proposed in this work. The goals of the evaluation are the following:

- Assess the performance of our sampling algorithm. In particular, to asses whether with probability $1-\delta$ the sets of frequent sequential patterns extracted from samples are ε-approximations, for the first strategy, and FPF ε-approximations, for the second one, of $FSP(\mathcal{D},\theta)$. In addition, we compared the performance of the sampling algorithm with the ones to mine the full datasets in term of execution time.
- Assess the performance of our algorithms for mining the true frequent sequential patterns. In particular, to assess whether with probability $1-\delta$ the set of frequent sequential patterns extracted from the dataset with the corrected threshold does not contain false positives, that is, it is a FPF μ-approximation of $TSFP(\pi,\theta)$, for the first method, and contains all the TFSPs, that is, it is a μ-approximation of $TSFP(\pi,\theta)$, for the second method. In addition, we compared the results obtained with the VC-dimension and with the Rademacher complexity, both used to compute an upper bound on the maximum deviation.

Since no sampling algorithm for rigorously approximating the set of frequent sequential patterns and no algorithm to mine true frequent sequential patterns have been previously proposed, we do not consider other methods in our experimental evaluation.

7.1. Implementation and Environment

The code to compute the bound on the VC-dimension (Algorithm 1) and to perform the evaluation has been developed in Java and executed using version 1.8.0_201. The code to compute the bound and the approximation to the Rademacher Complexity (resp. Algorithms 4 and 5) has been developed in C++. We have performed all our experiments on the same machine with 512 GB of RAM and 2 Intel(R) Xeon(R) CPU E5-2698 v3 @ 2.3GHz. To mine sequential patterns, we used the PrefixSpan [10] implementation provided by the SPMF library [28]. We used NLopt [29] as non-linear optimization solver. Our open-source implementation and the code developed for the tests, including scripts to reproduce all results, are available online [30].

7.2. Datasets

In this section, we describe the datasets we used in our evaluation. We first describe the dataset used to evaluate our sampling algorithm for FSP mining, and then the datasets used for TFSP mining. All datasets are obtained starting from the following real datasets:

- BIBLE: a conversion of the Bible into sequence where each word is an item;
- BMS1: contains sequences of click-stream data from the e-commerce website Gazelle;
- BMS2: contains sequences of click-stream data from the e-commerce website Gazelle;
- FIFA: contains sequences of click-stream data from the website of FIFA World Cup 98;
- KOSARAK: contains sequences of click-stream data from an Hungarian news portal;

- LEVIATHAN: is a conversion of the novel Leviathan by Thomas Hobbes (1651) as a sequence dataset where each word is an item;
- MSNBC: contains sequences of click-stream data from MSNBC website and each item represents the category of a web page;
- SIGN: contains sign language utterance.

All the datasets used are publicly available online [31] and the code to generate the pseudo-artificial datasets, as described in the following sections, is provided [30]. The characteristics of the datasets are reported in Table 1.

Table 1. Datasets characteristics. For each dataset $\mathcal{D}$, we report the number $|\mathcal{D}|$ of transactions, the total number $|\mathcal{I}|$ of items, the average transaction item-length and the maximum transaction item-length.

Dataset $\mathcal{D}$	**Size $|\mathcal{D}|$**	**$|\mathcal{I}|$**	**Avg. Item-Length**	**Max. Item-Length**
BIBLE	36,369	13,905	21.6	100
BMS1	59,601	497	2.5	267
BMS2	77,512	3340	4.6	161
FIFA	20,450	2990	36.2	100
KOSARAK	69,999	14,804	8.0	796
LEVIATHAN	5835	9025	33.8	100
MSNBC	989,818	17	4.8	14,795
SIGN	730	267	52.0	94

7.2.1. FSP Mining

The typical scenario for the application of sampling is that the dataset to mine is very large, sometimes even too large to fit in the main memory of the machine. Thus, in applying sampling techniques, we aim to reduce the size of such dataset, considering only a sample of it, in order to obtain an amount of data of reasonable size. Since the number of transactions in each real dataset (shown in Table 1) is fairly limited, we replicated each dataset to reach modern datasets sizes. For each real dataset, we fixed a replication factor and we created a new dataset, replicating each transaction in the dataset a number of times equal to the replication factor. Then, the input data for the sampling algorithm is the new enlarged dataset. The replication factors used are the following: BIBLE and FIFA = 200x; BMS1, BMS2 and KOSARAK = 100x; LEVIATHAN = 1000x; MSNBC = 10x and SIGN = 10,000x.

7.2.2. TFSP Mining

To evaluate our algorithms to mine the true frequent sequential patterns, we need to know which are the sequential patterns that are frequently generated from the unknown generative process π. In particular, we need a *ground truth* of the true frequencies of the sequential patterns. We generated pseudo-artificial datasets by taking some of the datasets in Table 1 as ground truth for the true frequencies t_π of the sequential patterns. For each ground truth, we created four new datasets by sampling sequential transactions uniformly at random from the original dataset. All the new datasets have the same number of transactions of the respectively ground truth, that is, the respectively original dataset. We used the original datasets as ground truth and we executed our evaluation in the new (sampled) datasets. Therefore, the true frequency of a sequential pattern is its frequency in the original dataset, that is, its frequency in the original dataset is exactly the same that such pattern would have in an hypothetical infinite number of transactions generated by the unknown generative process π.

7.3. Sampling Algorithm Results

In this section, we describe the results obtained with our sampling algorithm (Algorithm 6). As explained above, the typical scenario to apply sampling is that the dataset to mine is very large. Thus, we aim to reduce the size of such dataset, considering only a sample of it. In addition, from the sample, we aim to obtain a good approximation of the results that would have been obtained from the entire dataset. In all our experiments we fixed $\varepsilon = 0.01$ and $\delta = 0.1$. The steps of the evaluation are the following (Algorithm 6): given a dataset $\mathcal{D}_L$ as input, we compute the sample size m, using Algorithm 2, to obtain an $\varepsilon = 0.01$-approximation (resp. FPF 0.01-approximation) with probability at least $1 - \delta = 0.90$. Then, we extract a random sample S of m transactions from $\mathcal{D}_L$ and we run the algorithm to mine the frequent sequential patterns on S. Finally, we verify whether the set of frequent sequential patterns extracted from the sample is a 0.01-approximation (resp. FPS 0.01-approximation) to $FSP(\mathcal{D}_L, \theta)$. For each dataset $\mathcal{D}_L$ we repeat the experiment 5 times, and then we compute the fraction of times the sets of frequent sequential patterns extracted from the samples have the properties described in Definition 1 (resp. Definition 2). Table 2 shows the results.

Table 2. Sampling algorithms results. For each enlarged dataset $\mathcal{D}_L$, we report θ, the ratio $|S|/|\mathcal{D}_L|$ between the sample size $|S|$ and the size of the enlarged dataset $|\mathcal{D}_L|$, Max_Abs_Err, the maximum $\max_{p \in C_i} |f_{\mathcal{D}}(p) - f_{S_i}(p)|$, and Avg_Abs_Err, the average $\max_{p \in C_i} |f_{\mathcal{D}}(p) - f_{S_i}(p)|$, over the 5 samples S_i and with C_i the set of frequent sequential patterns extracted from S_i, the percentage of ε-approximations obtained over the 5 samples and the percentage of FPF ε-approximations obtained over the 5 samples.

Dataset $\mathcal{D}_L$	θ	$\lvert S\rvert/\lvert\mathcal{D}_L\rvert$	Max_Abs_Err ($\times 10^{-4}$)	Avg_Abs_Err ($\times 10^{-4}$)	ε-approx (%)	FPF ε-approx (%)
BIBLE	0.1	0.24	9.33	7.47	100	100
BMS1	0.012	0.17	5.45	4.70	100	100
BMS2	0.012	0.16	4.08	3.14	100	100
FIFA	0.25	0.50	8.68	7.07	100	100
KOSARAK	0.02	0.52	7.18	4.95	100	100
LEVIATHAN	0.15	0.30	9.19	7.84	100	100
MSNBC	0.02	0.37	4.33	3.63	100	100
SIGN	0.4	0.20	14.14	12.19	100	100

We observe that the samples obtained from the datasets are about 2 to 5 times smaller than the whole datasets. Moreover, in all the runs for all the datasets, we obtain an ε-approximation (resp. FPF ε-approximation). Such results are even better than the theoretical guarantees, that ensure to obtain such approximations with probability at least 90%. We also reported Max_Abs_Err $= \max_{S_i, i \in [1,5]} \max_{p \in C_i} |f_{\mathcal{D}}(p) - f_{S_i}(p)|$ and Avg_Abs_Err $= \frac{1}{5} \sum_{S_i, i \in [1,5]} \max_{p \in C_i} |f_{\mathcal{D}}(p) - f_{S_i}(p)|$, where C_i is the set of frequent sequential patterns extracted from the sample S_i, $i = 1, ..., 5$ (since we run each experiment 5 times, there are 5 samples). They represent the maximum and the average, over the 5 runs, of the maximum absolute difference between the frequency that the sequential patterns have in the entire dataset and that they have in the sample, over all the sequential patterns extracted from the sample. Again, the results obtained are better than the theoretical guarantees, that ensure a maximal absolute difference lower than $\varepsilon/2 = 0.005$.

Figure 3 shows the comparison between the average execution time of the sampling algorithm and the average execution time of the mining of the entire dataset, over the 5 runs. For all the datasets, the sampling algorithm requires less time than the mining of the whole dataset. For BMS1 and BMS2, the mining of the whole dataset is very fast since the number of frequent sequential patterns extracted from it is low. Thus, there is not a large difference between the execution time to mine the whole dataset and the execution time for the sampling algorithm, which is most due to the computation of the sample size. Similar results between our sampling algorithm and the mining of the whole dataset have also been obtained with KOSARAK and MSNBC. As expected, for all the datasets, the execution time of the sampling algorithm to obtain an ε-approximation is larger than the execution time of the sampling algorithm to obtain a FPF ε-approximation, since the minimum frequency threshold used in the first case is lower, resulting in a higher number of extracted sequential patterns.

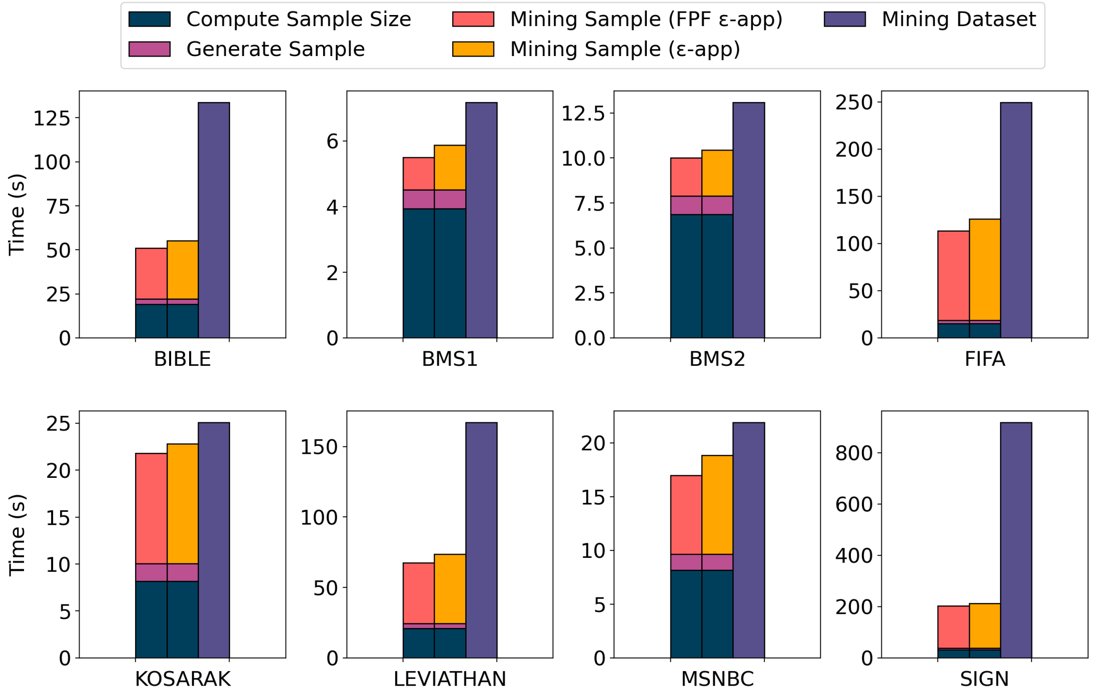


Figure 3. Execution time of the sampling algorithm. The execution time required to mine the whole dataset, and the execution times of the sampling algorithm to obtain an ε-approximation and a false positives free (FPF) ε-approximation are reported. For the sampling algorithms, we show the execution time to compute the sample size, the execution time to generate the sample, and the execution time to mine the sample.

We now discuss some of the patterns extracted from the MSNBC dataset, for which richer information regarding the data is available. In particular, in MSNBC each transaction contains the sequence of click-stream data generated by a single view on the MSNBC website by a user, and each item represents the category of a visited webpage, such "frontpage", "news", "sports", and so forth.

The two most frequent sequential patterns extracted in the enlarged datasets with a classic FSP algorithm are single categories, that is, sequential patterns of item-length 1: $\langle\{frontpage\}\rangle$ is the most frequent while $\langle\{on - air\}\rangle$ is the second one. They are also the two most frequent sequential patterns extracted in all the five samples using our sampling algorithms. The most frequent sequential patterns with item-length greater than one are the sequential patterns $\langle\{frontpage\},\{frontpage\}\rangle$ and $\langle\{frontpage\},\{frontpage\},\{frontpage\}\rangle$. For $\langle\{frontpage\},\{frontpage\}\rangle$, 75% of the transactions in which it appears there is at least an instance of such pattern where the two items are consecutive. This means that users visited two consecutive webpages of the same category, "frontpage", or that they refreshed the same page twice, while in the 25% of the transactions in which it appears users visited webpages of other categories between the two "frontpage" webpages. Instead, for $\langle\{frontpage\},\{frontpage\},\{frontpage\}\rangle$ the percentage of transactions in which the three items are consecutive is 59%. We also observed similar results with other categories: sequential patterns that are sequences of the same item, and so of the same category, have higher frequency. This fact highlights that users usually visit more frequently pages of the same category or that they refresh multiple times the same pages.

The most frequent sequential patterns that are not sequences of the same item are combinations of the items "frontpage" and "news", for example, $\langle\{frontpage\},\{news\}\rangle$, $\langle\{frontpage\},\{news\},\{news\}\rangle$ and $\langle\{news\},\{frontpage\}\rangle$. Surprisingly, the item "on-air" alone is more frequent that the item "news" alone. This means that users visit "news" webpages coming from a "frontpage" more frequently than "on-air" webpages, though they visit more frequently "on-air" webpages.

7.4. True Frequent Sequential Patterns Results

In this section, we describe the results of our algorithms for mining the true frequent sequential patterns. In all these experiments, we fixed $\delta = 0.1$. First of all, for each real dataset we generated 4 pseudo-artificial datasets $\mathcal{D}_i$, $i \in [1, 4]$ from the same ground truth. We mined the set $FSP(\mathcal{D}_i, \theta)$, and we compared it with the TFSPs, that is, the set $FSP(\mathcal{D}, \theta)$, where $\mathcal{D}$ is the ground truth. Such experiments aim to verify whether the sets of the FSPs extracted from the pseudo-artificial datasets contain false positives and miss some TFSPs. Table 3 shows the fractions of times that the set $FSP(\mathcal{D}_i, \theta)$ contains false positives and misses TFSPs from the ground truth. We ran this evaluation over the four datasets $\mathcal{D}_i$, $i \in [1, 4]$, of the same size from the same ground truth and we reported the average. For each dataset, we report the results with two frequency thresholds θ. In almost all the cases, the FSPs mined from the pseudo-artificial datasets contain false positives and miss some TFSPs. In particular, with lower frequency thresholds (and, therefore, a larger number of patterns), the fraction of times we find false positives and false negatives usually increases. These results emphasize that, in general, the mining of the FSPs is not enough to learn interesting features of the underlying generative process of the data, and techniques like the ones introduced in this work are necessary.

Table 3. Average fraction of times that $FSP(\mathcal{D}_i, \theta)$, with $\mathcal{D}_i$ a pseudo-artificial dataset, contains false positives, Times FPs, and misses true frequent sequential patterns (TFSPs) (false negatives), Times FNs, over 4 datasets $\mathcal{D}_i$ from the same ground truth.

Ground Truth	θ	|TFSP|	Times FPs	Times FNs
BIBLE	0.1	174	50%	100%
	0.05	774	100%	100%
BMS1	0.025	13	50%	0%
	0.0225	17	0%	25%
BMS2	0.025	10	0%	0%
	0.0225	11	0%	0%
KOSARAK	0.06	23	100%	0%
	0.04	41	50%	25%
LEVIATHAN	0.15	225	75%	100%
	0.1	651	100%	100%
MSNBC	0.02	97	75%	25%
	0.015	143	100%	50%

Then, we compute and compare the upper bounds to the maximum deviation introduced in the previous sections, since our strategy to find an approximation to the true frequent sequential patterns hinges on finding a tight upper bound to the maximum deviation. For each pseudo-artificial dataset, we computed the upper bound $\mu_{VC}/2$ to the maximum deviation using the VC-dimension based bound (ComputeMaxDevVC, Algorithm 3), the Rademacher complexity based bound $\mu_R^b/2$ (ComputeMaxDevRadeBound, Equation (34)), and the Rademacher complexity approximation $\mu_R^a/2$ (ComputeMaxDevRadeApprox, Equation (40)). Table 4 shows that the two methods for computing the upper bound to the maximum deviation using an upper to the empirical VC-dimension and Rademacher complexity are similar for BMS1 and BMS2, but for the other samples the VC-dimension-based algorithm is better than the one based on the Rademacher complexity bound by a factor between 2 and 3, that is, $\mu_R^b / \mu_{VC} \in [2, 3]$. Tighter upper bounds to the maximum deviation are provided by the method that uses the approximation of the Rademacher complexity.

Table 4. Comparison of the upper bound $\mu/2$ to the maximum deviation achieved respectively by ComputeMaxDevVC, ComputeMaxDevRadeBound, and ComputeMaxDevRadeApprox for each dataset. We show averages *avg*, maximum values *max*, and standard deviations *std* for each dataset and method over the 4 pseudo-artificial datasets.

Dataset	μ_{VC}**/2**			μ_R^b**/2**			μ_R^a**/2**		
	avg	**max**	**std ($\times 10^{-3}$)**	**avg**	**max**	**std ($\times 10^{-3}$)**	**avg**	**max**	**std ($\times 10^{-3}$)**
BIBLE	0.0339	0.0340	0.1	0.0747	0.0748	0.1	0.0207	0.0223	1.5
BMS1	0.0194	0.0197	0.3	0.0287	0.0294	0.6	0.0136	0.0153	1.0
BMS2	0.0194	0.0196	0.1	0.0202	0.0207	0.5	0.0107	0.0115	0.5
KOSARAK	0.0334	0.0335	0.1	0.0957	0.0972	1.5	0.0145	0.0164	1.5
LEVIATHAN	0.0847	0.0850	0.3	0.1878	0.1904	1.6	0.0569	0.0636	5.5
MSNBC	0.0089	0.0090	0.1	0.0252	0.0257	0.9	0.0035	0.0041	0.4

In our implementation of Algorithm 4 to compute an upper bound to the empirical Rademacher complexity of sequential patterns, we compute several upper bounds associated with different integer values of $\eta \in [\overline{\eta} + \beta_1, \min(\beta_2, \hat{\eta})]$ for fixed values of β_1 and β_2, taking the minimum bound among those computed. In our experiments, we fixed $\beta_1 = 20$ and $\beta_2 = 120$. In practice, by increasing the value of η we observe a decreasing trend of the upper bound value until a minimum value is reached. Then, by increasing again the value of η the value of the upper bound increases until it converges to the one achieved with $\eta = \hat{\eta}$. In addition, for each pseudo-artificial dataset the value of η associated with the minimum value of the upper bound to the maximum deviation is always found in $[\overline{\eta} + \beta_1, \min(\beta_2, \hat{\eta})]$, with $\beta_1 = 20$, $\beta_2 = 120$.

Finally, we evaluated the performance of our two strategies to mine an approximation of the true frequent sequential patterns, the first one with guarantees on the false positives and the second one with guarantees on the false negatives, using the upper bounds on the maximum deviation computed above. We considered the two tightest upper bounds, that are $\mu_{VC}/2$ and $\mu_R^a/2$, computed respectively using the empirical VC-dimension and an approximation of the empirical Rademacher complexity. From each pseudo-artificial dataset, we mined the FSPs using $\hat{\theta}$, for the first strategy, and $\tilde{\theta}$, for the second one, respectively computed using Theorems 10 and 11, and we compared the sequential patterns extracted with the TFSPs from the ground truth. Table 5 shows the results for the strategy with guarantees on the false positives. Using $\mu_{VC}/2$ to compute the corrected frequency threshold $\hat{\theta}_{VC}$, our algorithm performs better than the theoretical guarantees in all the runs, since the number of times the output contains false positives is always equal to zero, while the theory guarantees a probability of at least $1 - \delta = 0.9$ to obtain the correct approximation. Obviously, this also happens using $\mu_R^a/2$ to compute the corrected frequency threshold $\hat{\theta}_R$, since $\mu_{VC} > \mu_R^a$. We also computed the average fraction of TFSPs reported in the output by the algorithm, that is, $|FSP(\mathcal{D}_i, \hat{\theta})|/|TFSP|$, since we aim to obtain as many TFSPs as possible. For all the datasets, it is possible to notice that the results obtained with the Rademacher complexity are better than the ones obtained with the VC-dimension, since the Rademacher allows to obtain a higher percentage of TFSPs in output. Table 6 shows the results for the strategy with guarantees on the false negatives. Similar to the previous case, our algorithm performs better than the theoretical guarantees in all the runs, since the number of times the algorithm misses some TFSPs is always equal to zero, with both the VC-dimension and the Rademacher complexity based results. We also report the average fractions of patterns in the output that are TFSPs, that is, $|TFSP|/|FSP(\mathcal{D}_i, \tilde{\theta})|$, since we are interested in obtaining all the TFSPs but with less false positives as possible. Again, the results with the Rademacher complexity are better than the ones obtained with the VC-dimension, since the number of sequential patterns in the output of the algorithm that are TFSPs is higher using the Rademacher complexity.

Table 5. Results of our algorithm for the TFSPs with guarantees on the false positives in 4 pseudo-artificial datasets $\mathcal{D}_i$ for each ground truth. The table reports the frequency thresholds θ used in the experiments, the number of TFSPs in the ground truth, the number of times the output contains false positives using $\hat{\theta}_{VC} = \theta + \mu_{VC}/2$ as frequency threshold and the average fraction of the reported TFSPs in the output using such frequency threshold, the number of times the output contains false positives using $\hat{\theta}_R = \theta + \mu_R^a/2$ and the average fraction of the reported TFSPs in the output using such frequency threshold.

Ground Truth	θ	|TFSP|	Times FPs in FSP($\mathcal{D}_i, \hat{\theta}_{VC}$)	|FSP($\mathcal{D}_i, \hat{\theta}_{VC}$)|/ |TFSP|	Times FPs in FSP($\mathcal{D}_i, \hat{\theta}_R$)	|FSP($\mathcal{D}_i, \hat{\theta}_R$)|/ |TFSP|
BIBLE	0.1	174	0 %	0.55	0 %	0.68
	0.05	774	0 %	0.32	0 %	0.47
BMS1	0.025	13	0 %	0.38	0 %	0.48
	0.0025	17	0 %	0.29	0 %	0.43
BMS2	0.025	10	0 %	0.13	0 %	0.20
	0.0025	11	0 %	0.18	0 %	0.18
KOSARAK	0.06	23	0 %	0.41	0 %	0.73
	0.04	41	0 %	0.43	0 %	0.74
LEVIATHAN	0.15	225	0 %	0.30	0 %	0.41
	0.1	651	0 %	0.18	0 %	0.30
MSNBC	0.02	97	0 %	0.56	0 %	0.77
	0.015	143	0 %	0.50	0 %	0.76

Table 6. Results of our algorithm for the TFSPs with guarantees on the false negatives in 4 pseudo-artificial datasets $\mathcal{D}_i$ for each ground truth. The table reports the frequency thresholds θ used in the experiments, the number of TFSPs in the ground truth, the number of times the output of the algorithm misses some TFSPs using $\tilde{\theta}_{VC} = \theta - \mu_{VC}/2$ as frequency threshold and the average fraction of sequential patterns that are TFSPs in the output using such frequency threshold, the number of times the output of the algorithm misses some TFSPs using $\tilde{\theta}_R = \theta - \mu_R^a/2$ and the average fraction of sequential patterns that are TFSPs in the output using such frequency threshold.

Ground Truth	θ	|TFSP|	Times FNs in FSP($\mathcal{D}_i, \tilde{\theta}_{VC}$)	|TFSP|/ |FSP($\mathcal{D}_i, \tilde{\theta}_{VC}$)|	Times FNs in FSP($\mathcal{D}_i, \tilde{\theta}_R$)	|TFSP|/ |FSP($\mathcal{D}_i, \tilde{\theta}_R$)|
BIBLE	0.1	174	0 %	0.42	0 %	0.63
	0.05	774	0 %	0.09	0 %	0.33
BMS1	0.025	13	0 %	0.07	0 %	0.21
	0.0025	17	0 %	0.04	0 %	0.19
BMS2	0.025	10	0 %	0.03	0 %	0.32
	0.0025	11	0 %	0.01	0 %	0.19
KOSARAK	0.06	23	0 %	0.30	0 %	0.64
	0.04	41	0 %	0.04	0 %	0.49
LEVIATHAN	0.15	225	0 %	0.12	0 %	0.30
	0.1	651	0 %	0.01	0 %	0.13
MSNBC	0.02	97	0 %	0.42	0 %	0.77
	0.015	143	0 %	0.24	0 %	0.65

We now we briefly analyze the sequential patterns extracted from the MSNBC dataset using our TFSP algorithms. Since we considered the FSP extracted from the whole dataset as ground truth, that is, as TFSP, the considerations reported for the most frequent sequential patterns extracted from the whole dataset and from the samples (see previous section) are still valid for the true frequent sequential patterns that have higher frequency.

Using $\theta = 0.02$, as shown in Tables 5 and 6, we find 97 true frequent sequential patterns. In the four pseudo-artificial datasets we extracted on average ≈126 and ≈230 sequential patterns with guarantees on the false negatives, using respectively the approximation on the Rademacher complexity and the VC-dimension. With the algorithms with guarantees on the false positives, we mined ≈74 and ≈54 sequential patterns, respectively.

$\langle \{frontpage\}, \{frontpage\}, \{frontpage\}, \{frontpage\}, \{frontpage\}, \{frontpage\}, \{frontpage\} \rangle$ is the most frequent sequential pattern that is a TFSP but that it is not returned by our algorithm with guarantees on the false positives using the VC-dimension, that is, it is one of the allowed false negatives, in all the four pseudo-artificial datasets. Instead, the corresponding algorithm that uses the approximation of the Rademacher complexity always returned such sequential pattern as a TFSP. The most frequent sequential patterns that are true frequent but that are not returned by our algorithm with guarantees on the false positives using the approximation of the Rademacher complexity are $\langle \{frontpage\}, \{frontpage\}, \{news\}, \{news\} \rangle$ in two pseudo-artificial datasets, and $\langle \{frontpage\}, \{news\}, \{frontpage\}, \{frontpage\} \rangle$ and $\langle \{frontpage\}, \{news\}, \{frontpage\}, \{frontpage\} \rangle$ both in one pseudo-artificial dataset. Instead, the most frequent sequential patterns that are not true frequent but that are returned by our algorithms with guarantees on the false negatives, that is, they are some of the allowed false positives, are $\langle \{frontpage\}, \{on-air\}, \{on-air\} \rangle$, in three pseudo-artificial datasets and $\langle \{frontpage\}, \{local\}, \{frontpage\} \rangle$ in one, for both strategies.

8. Discussion

In this work, we studied two tasks related to sequential pattern mining: *frequent* sequential pattern mining and *true frequent* sequential pattern mining. For both tasks, we defined rigorous approximations and designed efficient algorithms to extract such approximations with high confidence using advanced tools from statistical learning theory. In particular, we devised an efficient sampling-based algorithm to approximate the set of frequent sequential patterns in large datasets using the concept of VC-dimension. We also devised efficient algorithms to mine the true frequent sequential patterns using VC-dimension and Rademacher complexity. Our extensive experimental evaluation shows that our sampling algorithm for mining frequent sequential patterns produces accurate approximations using samples that are small fractions of the whole datasets, thus vastly speeding up the sequential pattern mining task on very large datasets. For mining true frequent sequential patterns, our experimental evaluation shows that our algorithms obtain high-quality approximations, even better than guaranteed by their theoretical analysis. In addition, our evaluation shows that the upper bound on the maximum deviation computed using the approximation of the Rademacher complexity allows to obtain better results than the ones obtained with the upper bound on the maximum deviation computed using the empirical VC-dimension.

Author Contributions: Conceptualization, D.S., A.T., and F.V.; methodology, D.S., A.T., and F.V.; software, D.S. and A.T.; validation, D.S., A.T., and F.V.; formal analysis, D.S., A.T., and F.V.; investigation, D.S. and A.T.; resources, F.V.; data curation, D.S. and A.T.; writing—original draft preparation, D.S., A.T., and F.V.; writing—review and editing, D.S., A.T., and F.V.; visualization, D.S. and A.T.; supervision, F.V.; project administration, F.V.; funding acquisition, F.V. All authors have read and agreed to the published version of the manuscript.

Funding: Part of this work was supported by the University of Padova grant STARS: Algorithms for Inferential Data Mining, and by MIUR, the Italian Ministry of Education, University and Research, under PRIN Project n. 20174LF3T8 AHeAD (Efficient Algorithms for HArnessing Networked Data). and under the initiative "Departments of Excellence" (Law 232/2016).

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A. Missing Proofs

In this appendix we present the proofs not included in the main text.

Theorem 3. *Let S be a random sample of m transactions taken with replacement from the sequential dataset $\mathcal{D}$ and $\varepsilon, \delta \in (0,1)$. Let d be the s-bound of $\mathcal{D}$. If*

$$m \geq \frac{2}{\varepsilon^2}\left(d + \ln\frac{1}{\delta}\right), \tag{A1}$$

then $\sup_{p\in\mathbb{U}} |f_{\mathcal{D}}(p) - f_S(p)| \leq \varepsilon/2$ *with probability at least* $1 - \delta$.

Proof. From Theorem 1 in the main text we know that S is an $\varepsilon/2$-bag for $\mathcal{D}$ with probability at least $1 - \delta$. This means that for all $r \in \mathcal{R}$ we have

$$\left| \frac{|\mathcal{D} \cap r|}{|\mathcal{D}|} - \frac{|S \cap r|}{|S|} \right| \leq \frac{\varepsilon}{2}. \tag{A2}$$

Given a sequence $p \in \mathbb{U}$ and its support set $T_{\mathcal{D}}(p)$ on $\mathcal{D}$, that is the range r_p, and from the definition of range set of a sequential dataset, we have

$$\frac{|\mathcal{D} \cap r_p|}{|\mathcal{D}|} = f_{\mathcal{D}}(p) \tag{A3}$$

and

$$\frac{|S \cap r_p|}{|S|} = f_S(p). \tag{A4}$$

Thus, $\sup_{p\in\mathbb{U}} |f_{\mathcal{D}}(p) - f_S(p)| \leq \varepsilon/2$ with probability $\geq 1 - \delta$. □

Theorem 4. *Let* $\mathcal{D}$ *be a finite bag of* $|\mathcal{D}|$ *i.i.d. samples from an unknown probability distribution* π *on* $\mathbb{U}$ *and* $\delta \in (0, 1)$. *Let* d *be the s-bound of* $\mathcal{D}$. *If*

$$\mu_{VC} = \sqrt{\frac{2}{|\mathcal{D}|}\left(d + \ln\frac{1}{\delta}\right)}, \tag{A5}$$

then $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq \mu_{VC}/2$ *with probability at least* $1 - \delta$.

Proof. The proof is analogous to the proof of Theorem 3, when we consider the dataset $\mathcal{D}$ a random sample of a fixed size and we aim to compute an upper bound on the maximum deviation between the true frequency of a sequence and its frequency in $\mathcal{D}$. □

Lemma 1. *Consider a subset* W *of the dataset* $\mathcal{D}$, $W \subseteq \mathcal{D}$. *Let* $CS_W(\mathcal{D})$ *be the set of closed sequential patterns in* $\mathcal{D}$ *whose support set in* $\mathcal{D}$ *is* W, *that is,* $CS_W(\mathcal{D}) = \{p \in CS(\mathcal{D}) : T_{\mathcal{D}}(p) = W\}$, *with* $C = |CS_W(\mathcal{D})|$. *Then the number* C *of closed sequential patterns in* $\mathcal{D}$ *with* W *as support set satisfies:* $0 \leq C \leq |CS(\mathcal{D})|$.

Proof. The proof is organized in such a way: first, we show that the basic cases $C = 0$ and $C = 1$ hold, second, we prove the cases for $2 \leq C \leq |CS(\mathcal{D})|$.

Let us consider the case where W is a particular subset of $\mathcal{D}$ for which no sequence has W as support set in $\mathcal{D}$. Thus, $CS_W(\mathcal{D})$ is an empty set and $C = 0$. The case $C = 1$ is trivial, since it could happen that only one closed sequential pattern has W as support set in $\mathcal{D}$.

Now, before proving the cases for a generic value of C in $[2, \ldots, |CS(\mathcal{D})|]$, we start by considering the case $C = 2$. Let p_1,p_2 be two sequences with W as support set. Assume that each super-sequence of p_1 but not of p_2 has support lower than the support of p_1, and each super-sequence of p_2 but not of p_1 has support lower than the support of p_2. Now, let us focus on super-sequences of both p_1 and p_2. Let $\tau \in W$ be a transaction of W. We define $\mathbf{y}_\tau = \tau_{p_1,p_2}$ as the subsequence of τ restricted to only the sequences p_1 and p_2, preserving the relative order of their itemsets. For instance, let $p_1 = \langle A, B\rangle$, $p_2 = \langle C, D\rangle$ and $\tau = \langle A, C, F, D, B\rangle$, where A, B, C, D, F are itemsets: thus, $\mathbf{y}_\tau = \langle A, C, D, B\rangle$. Now, if the support set of $\mathbf{y}_\tau$ in W does not coincide with W, that is, $T_W(\mathbf{y}_\tau) \subset W$, then for each transaction $\tau \in W$ we have $|T_W(\mathbf{y}_\tau)| < |T_W(p_1)| = |T_W(p_2)| = |W|$. Note that this could happen because the set of itemsets of p_1 and p_2 may not appear in the same order in all transactions. Hence each super-sequence of both p_1 and p_2 has support lower than the support of p_1 (that is equal to the support of p_2). Thus, each super-sequence of p_i has a lower support compared to the support of p_i, for $i = 1, 2$.

This implies that p_1 and p_2 are closed sequences in $\mathcal{D}$ and since their support set is W, they belong to $CS_W(\mathcal{D})$. Thus, the case $C = 2$ could happen.

Now we generalize this concept for a generic number C of closed sequential patterns, where $2 \leq C \leq |CS(\mathcal{D})|$. Let $H = \{p_1, p_2, \ldots, p_C\}$ be a set of C sequential patterns with W as support set. Assume that each super-sequence of p_i but not of p_k has support lower than the support of p_i, for each $i, k \in [1, \ldots, C]$ with $k \neq i$. Let H_p be the power set of H without the empty set and the sets made of only one sequence, that is, $H_p = P(H) \setminus \{\{\emptyset\}, \{p_1\}, \{p_2\}, \ldots, \{p_C\}\}$. So, in H_p there are every possible subset of H of size greater than one. For a transaction $\tau \in W$ and $h_p \in H_p$, we define $\mathbf{y}_\tau(h_p) = \tau_{h_p}$ as the subsequence of τ restricted to h_p, that is, to only the sequences $p \in h_p$, preserving the relative order of their itemsets. If $\forall h_p \in H_p$ there exits a transaction $\tau \in W$ such that the support set of $\mathbf{y}_\tau(h_p)$ in W does not coincide with W, that is, $T_W(\mathbf{y}_\tau(h_p)) \subset W$, then for each transaction $\tau \in W$ we have $|T_W(\mathbf{y}_\tau(h_p))| < |T_W(p_1)| = \cdots = |T_W(p_C)| = |W|$. Hence each super-sequence made of only sequences of h_p has support lower than the support of p_i, for $i = 1, \ldots, C$. Thus, each super-sequence of p_i has a lower support compared to the support of p_i, for $i = 1, \ldots, C$. This implies that all sequences of H are closed sequence in $\mathcal{D}$ and since their support set is W, they belong to $CS_W(\mathcal{D})$. □

Lemma 2. *$V_\mathcal{D} = \{v_\mathcal{D}(p) : p \in CS(\mathcal{D})\} \cup \{(0, \ldots, 0)\}$ and $|V_\mathcal{D}| \leq |CS(\mathcal{D})| + 1$, that is, each vector of $V_\mathcal{D}$ different from $(0, \ldots, 0)$ is associated with at least one closed sequential pattern in $\mathcal{D}$.*

Proof. Let $V_\mathcal{D} = \overline{V}_\mathcal{D} \cup \{(0, \ldots, 0)\}$, where $\overline{V}_\mathcal{D} = \{v \in V_\mathcal{D} : v \neq (0, \ldots, 0)\}$. Let $p \in \mathbb{U}$ be a sequence of non-empty support set in $\mathcal{D}$, that is, $v_\mathcal{D}(p) \neq (0, \ldots, 0)$. There are two possibilities: p is or is not a closed sequence in $\mathcal{D}$. If p is not a closed sequence, then there exists a closed super-sequence $\mathbf{y} \sqsupset p$ with support equal to the support of p, so with $v_\mathcal{D}(p) = v_\mathcal{D}(\mathbf{y})$. Thus, $v_\mathcal{D}(p)$ is associated with at least one closed sequence. Combining this with the fact that each vector $v \in \overline{V}_\mathcal{D}$ is associated with at least one sequence $p \in \mathbb{U}$ and Lemma 1, then each vector of $V_\mathcal{D}$ different from $(0, \ldots, 0)$ is associated with at least one closed sequential pattern of $\mathcal{D}$. To conclude our proof is sufficient to show that there are no closed sequences associated with the vector $(0, \ldots, 0)$. Let $SP_\infty = \{p \in \mathbb{U} : v_\mathcal{D}(p) = (0, \ldots, 0)\}$. Note that $|SP_\infty| = \infty$. For each $p \in SP_\infty$, there always exists a super-sequence $\mathbf{y} \sqsupset p$ such that $f_\mathcal{D}(p) = f_\mathcal{D}(\mathbf{y}) = 0$. This implies that each sequence of SP_∞ is not closed. Thus, $\overline{V}_\mathcal{D} = \{v_\mathcal{D}(p) : p \in CS(\mathcal{D})\}$ and $|V_\mathcal{D}| = |\overline{V}_\mathcal{D}| + 1 \leq |CS(\mathcal{D})| + 1$. □

Lemma 3. *We have*

$$|C_{a,\tau}| \leq 2^{k_{a,\tau} - m_{a,\tau}}(2^{m_{a,\tau}} - 1). \tag{A6}$$

Proof. $C_{a,\tau}$ represents a subset of the set Φ of all those subsequences of τ that are made of only items equal to a or that come after a in $<_o$, with item-length at least two and with at least one occurrence of a. Let us focus on finding an upper bound to $|\Phi|$. In order to build such a generic subsequence of τ, it is sufficient to select i occurrences of a among the $m_{a,\tau}$ available, with $1 \leq i \leq m_{a,\tau}$, and choose j items among the remaining $k_{a,\tau} - m_{a,\tau}$ items different from a. Note that if $i = 1$, then j must be greater than 0. Thus, using the fact that the sum of $\binom{n}{k}$ for $k = 0, \ldots, n$ is equal to 2^n, we have

$$|\Phi| \leq \binom{m_{a,\tau}}{1} \sum_{j=1}^{k_{a,\tau} - m_{a,\tau}} \binom{k_{a,\tau} - m_{a,\tau}}{j} + \sum_{i=2}^{m_{a,\tau}} \left[\binom{m_{a,\tau}}{i} \sum_{j=0}^{k_{a,\tau} - m_{a,\tau}} \binom{k_{a,\tau} - m_{a,\tau}}{j} \right] \leq \tag{A7}$$

$$\leq 2^{k_{a,\tau} - m_{a,\tau}} \sum_{i=1}^{m_{a,\tau}} \binom{m_{a,\tau}}{i} = 2^{k_{a,\tau} - m_{a,\tau}}(2^{m_{a,\tau}} - 1), \tag{A8}$$

where the first inequality holds because some sequences of Φ are counted more times. Since $|C_{a,\tau}| \leq |\Phi|$, the thesis holds. □

Lemma 4. *Given an item a in $\mathcal{I}$, we define the following quantity:*

$$q(a,\eta) = 1 + \sum_{k=1}^{\chi_a} \sum_{m=1}^{k} \sum_{j=1}^{g_{a,k,m}} \left(\mathbb{1}(k \leq \eta) 2^{k-m}(2^m - 1) + \mathbb{1}(k > \eta) \sum_{i=1}^{\eta-1} \binom{k-1}{i} \right). \tag{A9}$$

Let $\tilde{w} : \mathbb{R}^+ \to \mathbb{R}^+$ be the function

$$\tilde{w}(s,\eta) = \frac{1}{s} \ln \sum_{a \in \mathcal{I}} \left(q(a,\eta) e^{\frac{s^2 f_{\mathcal{D}}(\langle\{a\}\rangle)}{2|\mathcal{D}|}} + |V_{\mathcal{D}(\eta)}| e^{\frac{s^2 |\mathcal{D}(\eta)|}{2|\mathcal{D}|^2}} + 1 \right). \tag{A10}$$

Then,

$$R_{\mathcal{D}} \leq \min_{s \in \mathbb{R}^+, \overline{\eta} < \eta \leq \hat{\eta}} \tilde{w}(s,\eta). \tag{A11}$$

Proof. Let us consider the function w from Theorem 7. For a given value of η, we have that $V_{\mathcal{D}} \subseteq (V_{\mathcal{D}} \setminus V_{\mathcal{D}(\eta)}) \cup V_{\mathcal{D}(\eta)}$, since not all the binary vectors of $V_{\mathcal{D}(\eta)}$ necessarily belong to $V_{\mathcal{D}}$. Thus:

$$w(s) = \frac{1}{s} \ln \sum_{v \in V_{\mathcal{D}}} \exp\left(\frac{s^2 ||v||^2}{2n^2}\right) \leq \frac{1}{s} \ln \left(\sum_{v \in V_{\mathcal{D}} \setminus V_{\mathcal{D}(\eta)}} \exp\left(\frac{s^2 ||v||^2}{2n^2}\right) + \sum_{v \in V_{\mathcal{D}(\eta)}} \exp\left(\frac{s^2 ||v||^2}{2n^2}\right) \right), \tag{A12}$$

where $n = |\mathcal{D}|$. For each binary vector $v \in V_{\mathcal{D}(\eta)}$ the maximum number of 1's is $\mid \mathcal{D}(\eta) \mid$. Thus,

$$\sum_{v \in V_{\mathcal{D}(\eta)}} \exp\left(\frac{s^2 ||v||^2}{2n^2}\right) \leq |V_{\mathcal{D}(\eta)}| \exp\left(\frac{s^2 |\mathcal{D}(\eta)|}{2n^2}\right). \tag{A13}$$

By using the definition of Euclidean norm, we have that, for any sequence $p \in \mathbb{U}$,

$$||v_{\mathcal{D}}(p)|| = \sqrt{\sum_{i=1}^{n} \phi_p(t_i)^2} = \sqrt{n f_{\mathcal{D}}(p)}. \tag{A14}$$

Note that each closed sequential pattern p with $||p|| > \eta$ can only appear in transactions of $\mathcal{D}(\eta)$ and, consequently, it is associated with a binary vector of $V_{\mathcal{D}(\eta)}$ and not of $V_{\mathcal{D}} \setminus V_{\mathcal{D}(\eta)}$. Thus, defining $CS(\mathcal{D},\eta)$ as the set of closed sequential patterns of $\mathcal{D}$ with item-length lower or equal to η and using Lemma 2 we can use the sum over $CS(\mathcal{D},\eta)$ as an upper bound on the sum over $V_{\mathcal{D}} \setminus V_{\mathcal{D}(\eta)}$:

$$\sum_{v \in V_{\mathcal{D}} \setminus V_{\mathcal{D}(\eta)}} \exp\left(\frac{s^2 ||v||^2}{2n^2}\right) \leq \sum_{p \in CS(\mathcal{D},\eta)} \exp\left(\frac{s^2 f_{\mathcal{D}}(p)}{2n}\right) + 1. \tag{A15}$$

Note that the vector $(0,\ldots,0)$ of $V_{\mathcal{D}} \setminus V_{\mathcal{D}(\eta)}$ provides a $+1$.

Now let us focus on the first term of the sum. The set $CS(\mathcal{D},\eta)$ can be broken using the Equation 29 in the sum over C_1

$$\sum_{p \in C_1} \exp\left(\frac{s^2 f_{\mathcal{D}}(p)}{2n}\right) \tag{A16}$$

plus the sum over $C_{2^+}(\eta)$ (i.e., the set of closed sequential patterns with item-length in $[2,\eta]$)

$$\sum_{a \in \mathcal{I}} \sum_{\tau \in T_{\mathcal{D}}(\langle\{a\}\rangle)} \sum_{p \in C_{a,\tau}(\eta)} \exp\left(\frac{s^2 f_{\mathcal{D}}(p)}{2n}\right), \tag{A17}$$

where $C_{a,\tau}(\eta)$ is the set of closed sequential patterns of $C_{a,\tau}$ with item-length in $[2, \eta]$. Since the set of items of the sequences in C_1 is a subset of $\mathcal{I}$, we have

$$\sum_{p \in C_1} \exp\left(\frac{s^2 f_{\mathcal{D}}(p)}{2n}\right) \leq \sum_{a \in \mathcal{I}} \exp\left(\frac{s^2 f_{\mathcal{D}}(\langle\{a\}\rangle)}{2n}\right). \tag{A18}$$

For any $p \in C_{a,\tau}(\eta)$, $f_{\mathcal{D}}(p) \leq f_{\mathcal{D}}(\langle\{a\}\rangle)$ by the anti-monotonicity support property for sequential patterns. An upper bound to the size of $C_{a,\tau}(\eta)$ can be computed in two ways, depending on the value of $k_{a,\tau}$. If $k_{a,\tau} \leq \eta$, we can use Lemma 3:

$$\sum_{\tau \in T_{\mathcal{D}}(\langle\{a\}\rangle)} \sum_{p \in C_{a,\tau}(\eta)} \exp\left(\frac{s^2 f_{\mathcal{D}}(p)}{2n}\right) \leq \sum_{\tau \in T_{\mathcal{D}}(\langle\{a\}\rangle)} 2^{k_{a,\tau} - m_{a,\tau}} (2^{m_{a,\tau}} - 1) \exp\left(\frac{s^2 f_{\mathcal{D}}(\langle\{a\}\rangle)}{2n}\right). \tag{A19}$$

If $k_{a,\tau} > \eta$ we have to count the number of possible closed sequential patterns with at least one item equal to a and with item-length in $[2, \eta]$ that we can build from $k_{a,\tau}$ items of τ:

$$\sum_{\tau \in T_{\mathcal{D}}(\langle\{a\}\rangle)} \sum_{p \in C_{a,\tau}(\eta)} \exp\left(\frac{s^2 f_{\mathcal{D}}(p)}{2n}\right) \leq \sum_{\tau \in T_{\mathcal{D}}(\langle\{a\}\rangle)} \sum_{i=1}^{\eta-1} \binom{k_{a,\tau} - 1}{i} \exp\left(\frac{s^2 f_{\mathcal{D}}(\langle\{a\}\rangle)}{2n}\right). \tag{A20}$$

Finally, using the quantities χ,k,m and g previously defined and indicator functions we can merge the right-hand sides of the last two inequalities

$$\sum_{k=1}^{\chi_a} \sum_{m=1}^{k} \sum_{j=1}^{g_{a,k,m}} (\mathbb{1}(k \leq \eta) 2^{k-m} (2^m - 1) + \mathbb{1}(k > \eta) \sum_{i=1}^{\eta-1} \binom{k-1}{i}) \exp\left(\frac{s^2 f_{\mathcal{D}}(\langle\{a\}\rangle)}{2n}\right). \tag{A21}$$

Thus, rearranging all the terms we reach the definition of $\tilde{w}$. Using the above arguments and the best value of η which minimizes the function we have that $w(s) \leq \tilde{w}(s, \eta)$ for any $s \in \mathbb{R}^+, \overline{\eta} < \eta \leq \hat{\eta}$. Since $R_{\mathcal{D}} \leq \min_{s \in \mathbb{R}^+} w(s)$ (by Theorem 7), we conclude that $R_{\mathcal{D}} \leq \min_{s \in \mathbb{R}^+, \overline{\eta} < \eta \leq \hat{\eta}} \tilde{w}(s, \eta)$. □

References

1. Agrawal, R.; Srikant, R. Mining sequential patterns. In Proceedings of the Eleventh International Conference on Data Engineering, Taipei, China, 6–10 March 1995; pp. 3–14.
2. Vapnik, V.N.; Chervonenkis, A.Y. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. In *Measures of Complexity*; Vovk, V., Papadopoulos, H., Gammerman, A., Eds.; Springer: Cham, Switzerland, 2015.
3. Boucheron, S.; Bousquet, O.; Lugosi, G. Theory of classification: A survey of some recent advances. *ESAIM Probab. Stat.* **2005**, *9*, 323–375. [CrossRef]
4. Riondato, M.; Upfal, E. Efficient discovery of association rules and frequent itemsets through sampling with tight performance guarantees. *ACM Trans. Knowl. Discov. D* **2014**, *8*, 20. [CrossRef]
5. Riondato, M.; Upfal, E. Mining frequent itemsets through progressive sampling with rademacher averages. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 22–27 August 2015; pp. 1005–1014.
6. Raïssi, C.; Poncelet, P. Sampling for sequential pattern mining: From static databases to data streams. In Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, USA, 28–31 October 2007; pp. 631–636.
7. Riondato, M.; Vandin, F. Finding the true frequent itemsets. In Proceedings of the 2014 SIAM International Conference on Data Mining, Philadelphia, PA, USA, 28 April 2014; pp. 497–505.
8. Servan-Schreiber, S.; Riondato, M.; Zgraggen, E. ProSecCo: Progressive sequence mining with convergence guarantees. *Knowl. Inf. Syst.* **2020**, *62*, 1313–1340. [CrossRef]

9. Srikant, R.; Agrawal, R. Mining sequential patterns: Generalizations and performance improvements. In *Advances in Database Technology–EDBT '96, Proceedings of the International Conference on Extending Database Technology, Avignon, France, 25–29 March 1996*; Springer: Berlin/Heidelberg, Germany, 1996; pp. 1–17.
10. Pei, J.; Han, J.; Mortazavi-Asl, B.; Wang, J.; Pinto, H.; Chen, Q.; Dayal, U.; Hsu, M.C. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1424–1440.
11. Wang, J.; Han, J.; Li, C. Frequent closed sequence mining without candidate maintenance. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 1042–1056. [CrossRef]
12. Pellegrina, L.; Pizzi, C.; Vandin, F. Fast Approximation of Frequent k-mers and Applications to Metagenomics. *J. Comput. Biol.* **2019**, *27*, 534–549. [CrossRef] [PubMed]
13. Riondato, M.; Vandin, F. MiSoSouP: Mining interesting subgroups with sampling and pseudodimension. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19 July 2018; pp. 2130–2139.
14. Al Hasan, M.; Chaoji, V.; Salem, S.; Besson, J.; Zaki, M.J. Origami: Mining representative orthogonal graph patterns. In Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, USA, 28–31 October 2007; pp. 153–162.
15. Corizzo, R.; Pio, G.; Ceci, M.; Malerba, D. DENCAST: distributed density-based clustering for multi-target regression. *J. Big Data* **2019**, *6*, 43. [CrossRef]
16. Cheng, J.; Fu, A.W.c.; Liu, J. K-isomorphism: privacy preserving network publication against structural attacks. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, Indianapolis, Indiana, 6–11 June 2010; pp. 459–470.
17. Riondato, M.; Upfal, E. ABRA: Approximating betweenness centrality in static and dynamic graphs with rademacher averages. *ACM Trans. Knowl. Discov. D* **2018**, *12*, 1–38. [CrossRef]
18. Mendes, L.F.; Ding, B.; Han, J. Stream sequential pattern mining with precise error bounds. In Proceedings of the Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 941–946.
19. Pellegrina, L.; Riondato, M.; Vandin, F. SPuManTE: Significant Pattern Mining with Unconditional Testing. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1528–1538.
20. Gwadera, R.; Crestani, F. Ranking Sequential Patterns with Respect to Significance. In *Advances in Knowledge Discovery and Data Mining*; Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V., Eds.; Springer: Berlin, Germany, 2010; Volume 6118.
21. Low-Kam, C.; Raïssi, C.; Kaytoue, M.; Pei, J. Mining statistically significant sequential patterns. In Proceedings of the IEEE 13th International Conference on Data Mining, Dallas, TX, USA, 7–10 December 2013; pp. 488–497.
22. Tonon, A.; Vandin, F. Permutation Strategies for Mining Significant Sequential Patterns. In Proceedings of the IEEE International Conference on Data Mining (ICDM), Beijing, China, 8–11 November 2019; pp. 1330–1335.
23. Mitzenmacher, M.; Upfal, E. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*; Cambridge University Press: New York, NY, USA, 2017.
24. Löffler, M.; Phillips, J.M. Shape fitting on point sets with probability distributions. In *Algorithms–ESA 2009, Proceedings of the European Symposium on Algorithms, Copenhagen, Denmark, 7–9 September 2009*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 313–324.
25. Li, Y.; Long, P.M.; Srinivasan, A. Improved bounds on the sample complexity of learning. *J. Comput. Syst. Sci.* **2001**, *62*, 516–527. [CrossRef]
26. Shalev-Shwartz, S.; Ben-David, S. *Understanding machine learning: From theory to algorithms*; Cambridge University Press: New York, NY, USA, 2014.
27. Egho, E.; Raïssi, C.; Calders, T.; Jay, N.; Napoli, A. On measuring similarity for sequences of itemsets. *Data Min. Knowl. Discov.* **2015**, *29*, 732–764. [CrossRef]
28. Fournier-Viger, P.; Lin, J.C.W.; Gomariz, A.; Gueniche, T.; Soltani, A.; Deng, Z.; Lam, H.T. The SPMF open-source data mining library version 2. In *Machine Learning and Knowledge Discovery in Databases*; Berendt, B., Ed.; Springer: Cham, Switzerland, 2016; Volume 9853, pp. 36–40.
29. Johnson, S.G. The NLopt Nonlinear-Optimization Package. 2014. Available online: https://nlopt.readthedocs.io/en/latest/ (accessed on 10 April 2020).

30. GitHub. VCRadSPM: Mining Sequential Patterns with VC-Dimension and Rademacher Complexity. Available online: https://github.com/VandinLab/VCRadSPM (accessed on 10 April 2020).
31. SPMF Datasets. Available online: https://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php (accessed on 10 April 2020).