2023 IEEE 39th International Conference on Data Engineering (ICDE) | 979-8-3503-2227-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDE55515.2023.00190

VC-dimension and Rademacher Averages of Subgraphs, with Applications to Graph Mining

Paolo Pellizzoni
University of Padova
paolo.pellizzoni@studenti.unipd.it

Fabio Vandin
University of Padova
fabio.vandin@unipd.it

***Abstract*—Frequent subgraph mining is a fundamental task in the analysis of collections of graphs. While several exact approaches have been proposed, it remains computationally challenging on large graph datasets due to its inherent link to the subgraph isomorphism problem and the huge number of candidate patterns even for fairly small subgraphs.**

In this work, we study two statistical learning measures of complexity, VC-dimension and Rademacher averages, for subgraphs, and derive efficiently computable bounds for both. We show how such bounds can be applied to devise efficient sampling-based approaches for rigorously approximating the solution of the frequent subgraph mining problem. We also show that our bounds can be used for *true* frequent subgraph mining, which requires to identify subgraphs generated with probability above a given threshold from an unknown generative process using samples from such process. Our extensive experimental evaluation on real datasets shows that our bounds lead to efficiently computable, high-quality approximations for both applications.

***Index Terms*—Frequent subgraphs, sampling, VC-dimension, Rademacher averages**

I. Introduction

Frequent subgraph mining is a fundamental data mining task that requires to identify small connected subgraphs appearing frequently in a collection of graphs. It finds applications in a large number of domains, including social media marketing [8], graph classification and clustering [7, 9], recommendation systems for video games [3], and computational biology [15].

The discovery of frequent subgraphs is a challenging task, due mostly to two reasons. First, to assess whether a subgraph appears in a graph requires to solve the subgraph isomorphism problem, which is, in general, NP-complete. Second, the number of candidate subgraphs is huge even for relatively small patterns. As a consequence, a number of exact approaches have been proposed [4, 10, 12, 16, 31, 34] . However, such exact approaches do not scale to large datasets.

Random sampling is a simple yet powerful technique to speed-up pattern mining, which has been applied to obtain *approximate* solutions for several patterns, such as itemsets [21], subgroups [25], and sequential patterns [26]. The major challenge in sampling approaches is to rigorously relate the results obtained from the analysis of the sample with the results that would be obtained analyzing the whole dataset, identifying sample sizes that provide rigorous guarantees on the quality of the approximation obtained from analyzing the sample.

Recently, advanced tools from statistical learning theory, such as the Vapnik-Chervonenkis (VC) dimension [30] and Rademacher averages [11], have been successfully used to obtain meaningful sample sizes required by random sampling for several pattern mining tasks [17, 21, 22, 25, 26, 27]. These tools improve over the use of standard approaches (e.g., using a Chernoff bound for a single pattern followed by a union bound on all patterns), which often lead to sample sizes that are *larger* than the original dataset. To the best of our knowledge, tools from statistical learning theory have not been applied to mining frequent subgraphs from a collection of graphs.

Contributions: In this paper we study the VC-dimension and Rademacher averages of subgraph patterns, and their use in sampling approaches for mining frequent subgraphs with guarantees. In particular, our contributions are four-fold. First, we derive rigorous bounds on the VC-dimension of the class of subgraph patterns that are efficiently computable for large datasets, improving the results that can be obtained by adapting the techniques previously proposes for other types of patterns. Second, we prove rigorous bounds on Rademacher averages for subgraph patterns, deriving different bounds specifically tailored for labelled graphs and for unlabelled graphs. Third, we show how our bounds on the VC-dimension and on the Rademacher average can be used to design effective algorithms to obtain approximate solutions with rigorous guarantees for two applications: frequent subgraph mining, and true frequent subgraph mining, a variant that requires to identify the subgraphs appearing with probability greater than a given threshold in a generative process having as input a set of samples from the process. Fourth, we perform an extensive experimental evaluation on real datasets, showing that the algorithms derived using our bounds obtain high-quality approximations for both applications, allowing the analysis of large collections of graphs in a fraction of the time required by exact approaches.

A. Related work

There is a huge literature on subgraph mining and related problems. Due to space limitations, we consider only the works most related to ours, focusing on the case of (static) collections of graphs, and refer the reader to the recent tutorial [28] for a more comprehensive overview of the field. Several advanced and finely-tuned algorithms [4, 10, 12,

2375-026X/23/$31.00 ©2023 IEEE
DOI 10.1109/ICDE55515.2023.00190

16, 31, 34] have been proposed for the exact solution of the frequent subgraph mining problem. However, all these approaches do not scale to large datasets. On the other hand, to the best of our knowledge no sampling approaches with rigorous guarantees have been proposed to approximate the set of frequent subgraphs in a collection of graphs. Our algorithms employ exact frequent subgraph miners as subroutines, but greatly reduce the required computational resources by analyzing a small sample of the dataset.

Statistical learning techniques such as VC dimension and Rademacher averages have been used to solve several pattern mining tasks, including itemset mining [17, 21, 22], sequential pattern mining [26, 27], and the computation of betweenness centrality [18, 20, 23]. Recently, Preti et al. [19] studied the VC-dimension of subgraph patterns of a single graph, focusing on the Minimum Node Image (MNI) frequency measure for subgraphs. The techniques from [19] cannot be easily adapted to our scenario of a collection of graphs and its related notion of frequency for subgraphs.

B. Roadmap

The paper is organized as follows. Section II provides all the preliminaries necessary for the rest of the paper. Section III presents an efficiently computable bound on the VC dimension of subgraphs. Section IV describes a novel bound on Rademacher averages of frequent subgraphs. Section V presents two applications of our novel bounds, for the tasks of frequent subgraph mining (Section V-A) and of true frequent subgraph mining (Section V-B). Section VI describes the experimental evaluation on several real-world datasets for both applications.

II. Preliminaries

In this section we provide the preliminary notions and results used in the rest of the paper. Section II-A describes the general setting of subgraph mining and useful approximations. Section II-B describes VC-dimension and its relation to the approximation of the frequency of patterns. Section II-C describes empirical VC-dimension, which generalizes VC-dimension to unknown distributions. Finally, Section II-D describes Rademacher averages and their relation to approximations of frequency of patterns.

A. Frequent Subgraph Mining, True Frequent Subgraph Mining, and Approximations

We define a dataset $\mathcal{D}$ as a collection of unweighted, undirected and labelled graphs $G = (V, E, L_V, L_E)$, where L_V and L_E are functions that map nodes and edges respectively to fixed labels. We refer to such graphs as *transactions* of the dataset. Two graphs G and G' are isomorphic if there exists a bijection μ from the nodes of the first one to the ones of the second one that preserves the node labels and s.t. $(u, v) \in E \iff (\mu(u), \mu(v)) \in E'$ and $L_E(u, v) = L_E(\mu(u), \mu(v))$. We say that graph H is isomorphic to an induced subgraph of G if there is an induced subgraph of G isomorphic to H. In $\mathcal{D}$ there can be isomorphic graphs.

Let $\mathcal{P}$ be the *pattern* set, that is a set of connected graphs whose frequency in the dataset is of interest. The set $\mathcal{P}$ can be the set of all connected graphs, the set of all connected graphs with up to k nodes, or a specific language of patterns of interest (although the mining algorithms have to be suitably modified for this last case). We say that G contains P, denoted by $P \subseteq G$, if P is isomorphic to an induced subgraph of G.

Given a pattern $P \in \mathcal{P}$, the support set $T_{\mathcal{D}}(P)$ of P is the set of transactions in $\mathcal{D}$ that contain the pattern P. The frequency of P is the fraction of transactions that contain P, $f_{\mathcal{D}}(P) = |T_{\mathcal{D}}(P)|/|\mathcal{D}|$.

Definition 1. *Given a dataset $\mathcal{D}$, a pattern set $\mathcal{P}$, and a frequency threshold θ, the* frequent subgraph mining *task is to find all patterns with frequency above θ along with their frequencies, that is*

$$FG(\mathcal{D}, \mathcal{P}, \theta) = \{(P, f_{\mathcal{D}}(P)) : \ P \in \mathcal{P} \text{ and } f_{\mathcal{D}}(P) \geq \theta\}$$

One often can sacrifice solving the above task exactly, and settle for an approximate solution (e.g., since often the threshold θ is chosen arbitrarily, or since the dataset is too large to obtain an exact solution in reasonable time). A commonly used notion of approximation for frequent pattern mining (e.g., [21, 26]), which guarantees no false negatives, is the following.

Definition 2. *Given a constant $\varepsilon > 0$, an absolute ε-close approximation to $FG(\mathcal{D}, \mathcal{P}, \theta)$ is a set $C = \{(P, \hat{f}(P)) : P \in \mathcal{P}$ and $\hat{f}(P) \in [0, 1]\}$ such that*

i) *$FG(\mathcal{D}, \mathcal{P}, \theta) \subseteq C$;*
ii) *C contains no pattern with frequency $f_{\mathcal{D}}(P) < \theta - \varepsilon$;*
iii) *for each $(P, \hat{f}(P))$ it holds $|\hat{f}(P) - f_{\mathcal{D}}(P)| \leq \varepsilon/2$.*

A simple yet effective idea to obtain absolute ε-close approximations for the frequent subgraph mining problem is to draw a random sample of the input dataset and to report frequent subgraphs appearing in the sample. As with all such sampling approaches, the main challenge is to identify a sample size that provably leads to a ε-close approximation.

Another problem we consider is the mining of *true frequent subgraphs*, which adapts the problem of true frequent itemsets mining [25] to subgraphs. In this scenario, the dataset $\mathcal{D}$ consists of a number of graphs obtained from an unknown generative process, described by an (unknown) generating distribution π, with $p_\pi(G)$ being the probability that graph G is the graph generated by π. The probability that a subgraph P is in a sample from the distribution π is then $p_\pi(P) = \sum_{G:P\subseteq G} p_\pi(G)$. The goal is then to identify the set $FG(\pi, \mathcal{P}, \theta)$ of all subgraphs P in a pattern set $\mathcal{P}$ with frequency $p_\pi(P) \geq \theta$, where θ is a minimum probability threshold. This problem is highly relevant when one is analyzing a dataset obtained from an underlying process and needs guarantees on the process itself (instead that on the dataset). Note that the true frequent subgraph mining problem cannot be solved exactly without full knowledge of π. For true frequent subgraph mining, the definition of ε-close approximation is a simple adaptation of Def. 2, obtained

by replacing $FG(\mathcal{D}, \mathcal{P}, \theta)$ with $FG(\pi, \mathcal{P}, \theta)$ and $f_{\mathcal{D}}(P)$ with $p_\pi(G)$.

B. Range Spaces and VC-dimension

The VC-dimension is a measure of the complexity of a family of indicator functions over a space of points, and it has been widely used in the context of machine learning.

Definition 3. *We define a range space as a pair (X, R) where X is a set and R is a family of subsets of X. The projection of R on $A \subseteq X$ is $P_R(A) = \{r \cap A : r \in R\}$. If $P_R(A) = 2^A$ we say that A is shattered by R. The* VC-dimension *$VC(X, R)$ of the range space is the largest cardinality of a subset of X which is shattered by R.*

Figure 1 shows an example of range space, where X is the set of points on a line, and R is the family of segments of the line. The VC dimension of such a range space is 2, since there exists a set of two points on a line that can be shattered by the family of segments, but there is no set of three points that can be shattered by it.

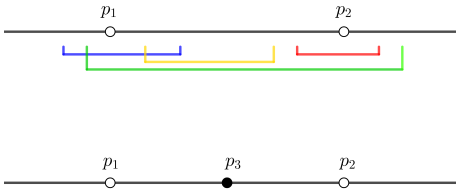

Fig. 1: The set $\{p_1, p_2\}$ can be shattered by the family of segments. On the other hand, any set of three points cannot be shattered, since it is impossible to include the extremal ones (p_1 and p_2 in the figure) in a segment without including the middle one (p_3 in the figure).

Given a range space, one can define a ε-approximation for a subset as follows.

Definition 4. *Let (X, R) be a range space and let A be a finite subset of X. For $0 < \varepsilon < 1$, a subset $B \subset A$ is an ε-approximation for A if for all $r \in R$ we have*

$$\left| \frac{|A \cap r|}{|A|} - \frac{|B \cap r|}{|B|} \right| \leq \varepsilon$$

The following result [29] provides a bound on the number of samples needed to obtain a ε-approximation of a range space as a function of the VC-dimension. Intuitively, the higher the VC dimension is, the more samples are needed to get an ε-approximation, due to the higher complexity of the range space.

Theorem 1. *There is an absolute positive constant c such that if (X, R) is a range space of VC dimension at most v, $A \subset X$ is a finite subset and $0 < \varepsilon, \delta < 1$, then a random subset (taken uniformly at random and with replacements) $B \subseteq A$ of cardinality $m = \min\left\{|A|, c/\varepsilon^2(v + \log(1/\delta))\right\}$ is an ε-approximation for A with probability $1 - \delta$.*

Thanks to some experimental evidence presented in [14], the constant c is usually considered to be close to 0.5.

C. Empirical VC-dimension

The VC-dimension, defined in the previous section, provides a bound on the number of samples to obtain a ε-approximation for a range space. However, it requires the computation of the VC-dimension of the entire range space. The empirical VC-dimension, defined below, considers instead the VC-dimension of a subset of the range space.

Definition 5. *Let (X, R) be a range space and let $Y \subset X$. Then the* empirical VC-dimension *of (X, R) on Y, denoted as $EVC((X, R), Y)$, is the VC dimension of the range space (Y, R'), with $R' = \{Y \cap r : \ r \in R\}$.*

The empirical VC-dimension allows to derive the accuracy provided by a random sample of a set *without* requiring to know the whole range space, as proved by the following result [21].

Lemma 1. *Let (X, R) be a range space and Y a random subset of X of size n. Let also $EVC((X, R), Y) \leq u$. Then, with probability $1 - \delta$, Y is an ε-approximation for X for*

$$\varepsilon = 2\sqrt{\frac{2u\ln(n+1)}{n}} + \sqrt{\frac{2\ln(2/\delta)}{n}}$$

D. Rademacher Averages

This section introduces Rademacher averages, another tool from statistical learning theory that allows to relate the frequencies observed in a sample with the frequencies in a full dataset, usually providing tighter bounds compared to the ones yielded by the empirical VC dimension. For ease of exposition, we introduce Rademacher averages referring directly to the functions of interests for subgraph mining.

For each pattern (i.e., subgraph) $P \in \mathcal{P}$, define the indicator function $\phi_P : \mathcal{D} \to \{0, 1\}$ as

$$\phi_P(G) = \begin{cases} 1 \text{ if } P \subseteq G \\ 0 \text{ otherwise} \end{cases}$$

Then we have $f_{\mathcal{D}}(P) = \frac{1}{|\mathcal{D}|} \sum_{G \in \mathcal{D}} \phi_P(G)$.

Assume to take a random sample (uniformly and with replacement) $\mathcal{S} = \{G_1, \ldots, G_n\}$ of size n from $\mathcal{D}$. Then let σ_i be a Rademacher random variable, i.e. taking value 1 or -1 with probability $1/2$, for each $1 \leq i \leq n$. Let also the σ_i's be independent. Then we define the following quantity:

Definition 6. *The (sample) conditional Rademacher average is*

$$\mathcal{R}_{\mathcal{S}} = \mathbb{E}_\sigma \left[\sup_{P \in \mathcal{P}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \phi_P(G_i) \right]$$

where $\mathbb{E}_\sigma$ denotes the expectation taken only with respect to the σ_i's, conditionally on the sample.

Then we have the following result [5] bounding the maximum deviation of pattern frequencies.

Theorem 2. *Let $\mathcal{S}$ be a random sample of a dataset $\mathcal{D}$ of cardinality n. With probability $1-\delta$,*

$$\sup_{P\in\mathcal{P}} |f_{\mathcal{D}}(G) - f_{\mathcal{S}}(G)| \leq 2\mathcal{R}_{\mathcal{S}} + \sqrt{\frac{2\ln 2/\delta}{n}}.$$

III. VC-DIMENSION OF SUBGRAPHS

In this section we derive new bounds on the VC-dimension of the range space of subgraphs in a dataset consisting of a collection of graphs. These bounds can be used for various subgraph mining tasks, as shown in Section V.

We start by defining the range space for subgraph mining.

Definition 7. *Let $\mathcal{D}$ be a dataset of transactions (i.e., graphs) and $\mathcal{P}$ a set of patterns (i.e., subgraphs). We define $(\mathcal{D}, R_{\mathcal{P}})$ as the range space such that $R_{\mathcal{P}} = \{r_P = T_{\mathcal{D}}(P) : P \in \mathcal{P}\}$ is a family of sets of transactions where, for each pattern $P \in \mathcal{P}, T_{\mathcal{D}}(P)$ is the set of graphs of $\mathcal{D}$ containing P.*

Figure 2 shows an example of a range space in the context of subgraphs. In this case, the range space $(\mathcal{D}, R_{\mathcal{P}})$ has VC-dimension 2, since the set of two graphs $\mathcal{A}$ can be shattered, but no set of three graphs from $\mathcal{D}$ can be shattered.

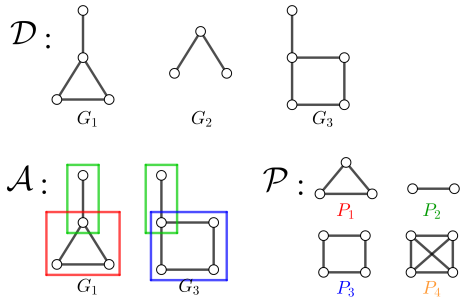

Fig. 2: An example of range space for subgraphs. The set $\mathcal{A}$ can be shattered by $\mathcal{P}$, since $T_{\mathcal{A}}(P_1) = \{G_1\}$, $T_{\mathcal{A}}(P_3) = \{G_3\}$, $T_{\mathcal{A}}(P_2) = \{G_1, G_3\}$ and $T_{\mathcal{A}}(P_4) = \emptyset$. On the other hand, no set of three graphs from $\mathcal{D}$ can be shattered, since the set $\{G_2\}$ cannot be obtained as a support set.

From the above definition we can clearly see that for a subset $\mathcal{D}'$ of $\mathcal{D}$, the quantity $\frac{|\mathcal{D}' \cap r_P|}{|\mathcal{D}'|} = \frac{|\mathcal{D}' \cap T_{\mathcal{D}}(P)|}{|\mathcal{D}'|}$ is $f_{\mathcal{D}'}(P)$, the frequency of P in $\mathcal{D}'$. The adaptation of techniques previously proposed for other types of patterns (e.g., itemsets [21] and sequences [26]) leads to upper bounds to the VC-dimension which can be computed with a total complexity of $O\left(|\mathcal{D}|^{2.5}\log|\mathcal{D}|\right)$. This complexity makes it impossible to compute such bounds on modern datasets, whose sizes easily exceed the millions of transactions.

We then present a new bound that can be computed in a single scan of the dataset. The following lemma is a simple adaptation of [21, Th. 4.2] to a dataset of graphs, and is used to prove our upper bound to the VC-dimension of subgraphs.

Lemma 2. *Let $\mathcal{D}$ be a dataset of transactions, $\mathcal{P}$ a set of patterns, and let $(\mathcal{D}, R_{\mathcal{P}})$ be the associated range space. Then $VC(\mathcal{D}, R_{\mathcal{P}}) \geq v$ if and only if there exists a set $\mathcal{A} \subseteq \mathcal{D}$ of v transactions such that for each subset $\mathcal{B} \subseteq \mathcal{A}$, there exists a pattern $P_{\mathcal{B}}$ such that the support set of $P_{\mathcal{B}}$ in $\mathcal{A}$ is exactly $\mathcal{B}$, i.e. $T_{\mathcal{A}}(P_{\mathcal{B}}) = \mathcal{B}$.*

Proof. If: We have that $T_{\mathcal{D}}(P_{\mathcal{B}}) \cap \mathcal{A} = \mathcal{B}$, for each of the $2^{|\mathcal{A}|}$ subsets $\mathcal{B}$ of $\mathcal{A}$. Hence $\mathcal{A}$ is shattered by $R_{\mathcal{P}}$ and $VC(\mathcal{D}, R_{\mathcal{P}}) \geq v$.

Only if: Let $VC(\mathcal{D}, R_{\mathcal{P}}) \geq v$. Then, there is a set $\mathcal{A} \subseteq \mathcal{D}$ of v transactions such that $P_{R_{\mathcal{P}}}(\mathcal{A}) = 2^{\mathcal{A}}$. Hence, for each subset $\mathcal{B}$ of $\mathcal{A}$, there exists a pattern $P_{\mathcal{B}}$ such that $T_{\mathcal{D}}(P_{\mathcal{B}}) \cap \mathcal{A} = \mathcal{B}$. □

We now define our novel upper bound to the VC-dimension of subgraphs, which we call c-bound.

Definition 8. *Let $\mathcal{D}$ be a dataset and $\mathcal{P}$ a set of patterns. The c-bound of $\mathcal{D}$ w.r.t. $\mathcal{P}$ is the maximum integer c such that $\mathcal{D}$ contains at least c different transactions G_i such that G_i contains at least 2^{c-1} distinct patterns from $\mathcal{P}$.*

The following proves that the c-bound is an upper bound to the VC-dimension of subgraphs.

Theorem 3. *Let $\mathcal{D}$ be a dataset with c-bound c w.r.t the pattern set $\mathcal{P}$. Then the range space $(\mathcal{D}, R_{\mathcal{P}})$ has VC-dimension $\leq c$.*

Proof. Let $l > c$ and assume that $(\mathcal{D}, R_{\mathcal{P}})$ has VC-dimension l. Then there is a set of $\mathcal{A}$ of l transactions of $\mathcal{D}$ that is shattered by $R_{\mathcal{P}}$, and by Lemma 2 for each subset $\mathcal{B} \subseteq \mathcal{A}$ there exists a pattern $P_{\mathcal{B}}$ such that $T_{\mathcal{A}}(P_{\mathcal{B}}) = \mathcal{B}$. Note that there must be a graph $G \in \mathcal{A}$ that contains at most $2^{(c+1)-1}-1$ patterns from $\mathcal{P}$, or $\mathcal{D}$ would have c-bound at least $c+1$.

This graph G is a member of 2^{l-1} subsets of $\mathcal{A}$, which we call $\mathcal{B}_i$'s, labelled in any order. Since $\mathcal{A}$ is shattered by $R_{\mathcal{P}}$, we have that for each set $\mathcal{B}_i$, there exists a pattern P_i such that $T(P_i) \cap \mathcal{A} = \mathcal{B}_i$. Note that since the $\mathcal{B}_i$'s are all different, then also the P_i's must be all different. Since $G \in \mathcal{B}_i$ by construction, we must have that $G \in T(P_i)\ \forall 1 \leq i \leq 2^{l-1}$. Then all such 2^{l-1} distinct patterns P_i appear as an induced subgraph of G. But, since $l > c$, we have that $2^{l-1} \geq 2^c > 2^c - 1$, and we have a contradiction. □

The condition "G_i contains at least 2^{c-1} distinct patterns from $\mathcal{P}$" is computationally difficult to test, since it requires to solve the subgraph isomorphism problem. Hence, we provide an algorithm to obtain an upper bound the c-bound of a dataset $\mathcal{D}$ in the case where $\mathcal{P}$ is the set of subgraphs with at most k nodes, which is a situation of interest in several subgraph mining applications.

Consider a graph $G \in \mathcal{D}$ with n_G nodes. Then G can contain at most $\hat{n}_G = \sum_{j=1}^{\min(k,n_G)} \min\left\{\binom{n_G}{j}, |\mathcal{P}_j|\right\}$ patterns from $\mathcal{P}$, where $|\mathcal{P}_j|$ is the number of patterns in $\mathcal{P}$ with j nodes. Let then $c_G = \lfloor \log_2(\hat{n}_G) \rfloor + 1$ and let $\hat{c}$ be the maximum integer c such that at least c transactions G have $c_G \geq c$. This value can be computed in a single pass over the dataset with Algorithm 1.

Note that, if the c-bound of $\mathcal{D}$ is c^*, then $c^* \leq \hat{c}$. Indeed, consider c^* graphs such that each one of them has at least 2^{c^*-1} distinct patterns from $\mathcal{P}$. Then for each of such graphs

Algorithm 1: COMPUTEBOUND($\mathcal{D}$)

$q = 0$
$T = \emptyset$
while HASNEXTTRANSACTION($\mathcal{D}$) **do**
  $G =$ NEXTTRANSACTION($\mathcal{D}$)
  $c_G = \left\lfloor \log \left(\sum_{j=1}^{\min(k, n_G)} \min \left\{ \binom{n_G}{j}, |\mathcal{P}_j| \right\} \right) \right\rfloor + 1$
  if $c_G > q$ *and* $\nexists H \in T$ *such that* $G = H$ **then**
    $T = T \cup \{G\}$
    if $\forall H \in T,\ c_H > q$ **then**
      $q++$
    else
      remove from T the transaction H with minimum c_H
return q

we have $2^{c^*-1} \leq \hat{n}_G$, so $c_G \geq c^*$. Then, since there are at least c^* graphs such that each one has $c_G \geq c^*$, we have $\hat{c} \geq c^*$. Hence, computing $\hat{c}$ provides an upper bound to the VC-dimension of $(\mathcal{D}, R_{\mathcal{P}})$.

Note that in general, although the bound might seem to be quite slack, the c-bound is in fact tight, in the sense that there exist datasets of graphs, both labeled and unlabeled, that have VC dimension exactly equal to their c-bound (proof in Section III-A). As a matter of fact, as shown in Section VI, on real-world datasets the bounds on the VC-dimension based on the c-bound are sharp enough to produce approximate frequent subgraph mining algorithms that reduce running times and peak memory consumption by two to three orders of magnitude compared to the exact methods.

A. Tightness

In this section we show that the c-bound is tight in the worst case. We first construct an example using labeled complete graphs based on the construction shown in [21]. In this example we use the fact that labeled graphs can be seen as a generalization of itemsets.

Theorem 4. *Let $\mathcal{P}$ be the set of all subgraphs. There exists a dataset $\mathcal{D}$ with c-bound c w.r.t. $\mathcal{P}$ such that the corresponding range space has VC-dimension c.*

Proof. Let the label set be $\mathbb{N}$ and let G_i be the complete graph on the c nodes labeled as $0, 1, \ldots, i-1, i+1, \ldots, c$. Let $\mathcal{A} = \{G_i : 1 \leq i \leq c\}$. $\mathcal{D}$ is a dataset containing $\mathcal{A}$ and any number of graphs with a single node labeled 0. Clearly the c-bound of $\mathcal{D}$ is c since there are the c graphs of $\mathcal{A}$ with $2^c - 1 > 2^{c-1}$ induced subgraphs each, but no $c+1$ graphs with at least 2^c induced subgraphs each.

Then for each $\mathcal{B} \in 2^{\mathcal{A}}$, $\mathcal{B} \neq \emptyset, \mathcal{A}$, let $P_{\mathcal{B}}$ be the complete graph on the nodes labeled with the numbers $\{1, \ldots, c\} \setminus \{i : G_i \in \mathcal{B}\}$. Then by construction $T_{\mathcal{A}}(P_{\mathcal{B}}) = \mathcal{B}$. Moreover the graph with a single node labeled 0 is contained in all graphs of $\mathcal{A}$ and the graph with a single node labeled $c+1$ in none. Then $\mathcal{A}$ is shattered and $VC(\mathcal{D}, \mathcal{R}_{\mathcal{P}}) \geq c$. □

One could argue that at least in the case where the graphs have few labels or are unlabeled the VC dimension of the range space should be smaller. Indeed, for frequent itemsets, if the ground set of the dataset has ℓ items, the VC dimension of the corresponding range space cannot exceed ℓ. We show, with a novel construction, that for graphs this is not in fact the case, and that the c-bound is tight even if we have only two labels.

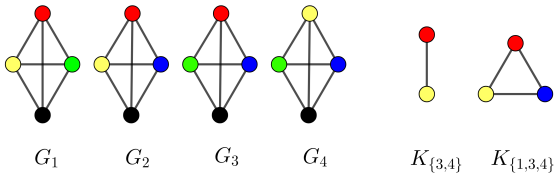

Fig. 3: An example of the construction for Theorem 4 for $c = 4$. We have that $T_{\mathcal{A}}(K_{\{1,\ldots,4\}\setminus\{1,2\}}) = \{G_1, G_2\}$ and that $T_{\mathcal{A}}(K_{\{1,\ldots,4\}\setminus\{2\}}) = \{G_2\}$

Theorem 5. *Let $\mathcal{P}$ be the set of all subgraphs. There exists a dataset $\mathcal{D}$ of graphs with only two labels with c-bound c w.r.t. $\mathcal{P}$ such that the corresponding range space has VC-dimension c.*

Proof. (sketch) Note that for a given c, we can find a set of c pairwise non-isomprphic unlabeled graphs $G_1, \ldots, G_c$ each with $k = O(\log c)$ nodes. We make use of the following gadget: for one such graph G_i, we label all of its nodes with 0 and connect all of them to a new node labeled 1. We call this new graph G'_i. The graph $G'_{\{i_1,\ldots,i_r\}}$ is the graph obtained by connecting all of the nodes of the graphs $G_{i_1}, \ldots, G_{i_r}$, labeled with 0, through a single node labeled with 1. Let also $G''_i = G'_{\{1,\ldots,i-1,i+1,c\}}$. Let then $\mathcal{A} = \{G''_i : 1 \leq i \leq c\}$. $\mathcal{D}$ is a dataset containing $\mathcal{A}$ and any number of graphs with a single node labeled 0. Clearly the c-bound of $\mathcal{D}$ is c since there are the c graphs of $\mathcal{A}$ with $2^c - 1 > 2^{c-1}$ distinct induced subgraphs each, but no $c+1$ graphs with at least 2^c induced subgraphs each.

We show that $\mathcal{A}$ can be shattered. Let $\mathcal{B} = \{G''_{i_1}, \ldots, G''_{i_r}\}$. We show that $T_{\mathcal{A}}(G'_{\{1,\ldots,c\}\setminus\{i_1,\ldots,i_r\}}) = \mathcal{B}$. Let $G''_{i_j} = G_{\{1,\ldots,i_j-1,i_j+1,\ldots,c\}} \in \mathcal{B}$. Then the pattern $G'_{\{1,\ldots,c\}\setminus\{i_1,\ldots,i_j,\ldots,i_r\}}$ appears in G''_{i_j}, as the only missing gadget in G''_{i_j} is G_{i_j}, which is missing also in the pattern $G'_{\{1,\ldots,c\}\setminus\{i_1,\ldots,i_j,\ldots,i_r\}}$. On the contrary, if $G''_{i_j} = G_{\{1,\ldots,i_j-1,i_j+1,\ldots,c\}} \notin \mathcal{B}$, then the pattern $G'_{\{1,\ldots,c\}\setminus\{i_1,\ldots,i_j,\ldots,i_r\}}$ cannot appear in G''_{i_j}, as in this pattern there is the gadget G_{i_j}, which does not appear in G''_{i_j}. Hence $\mathcal{A}$ is shattered and $VC(\mathcal{D}, \mathcal{R}_{\mathcal{P}}) \geq c$. □

With a little more care to avoid overlapping patterns one can construct an example where the c-bound is tight even with a single label.

IV. RADEMACHER AVERAGES OF SUBGRAPHS

In this section we show how to efficiently bound the Rademacher average $\mathcal{R}_{\mathcal{S}}$ of subgraphs. Indeed, note that computing $\mathcal{R}_{\mathcal{S}}$ directly is infeasible, as it involves computing the support of all patterns, and we hence resort to bounds

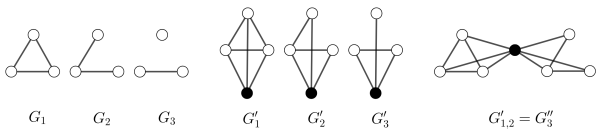

Fig. 4: An example of the gadgets for Theorem 5 for $c = 3$.

given by Massart's lemma [29], which is the most widely used tools to bound Rademacher averages without resorting to time-consuming Monte-Carlo methods.

Given a sample $\mathcal{S} = \{G_1, \ldots, G_n\}$ of n graphs, for any pattern (i.e., subgraph) $P \in \mathcal{P}$ let $v_P \in \mathbb{R}^n$ be the n-dimensional binary vector $(\phi_P(G_1), \ldots, \phi_P(G_n))^T$. Then, let $V_\mathcal{S} = \{v_P,\ P \in \mathcal{P}\}$. Since $V_\mathcal{S}$ is a set, we may have $|V_\mathcal{S}| \ll |\mathcal{P}|$ in case many patterns exhibit the same support in $\mathcal{S}$.

Theorem 6 (Massart's Lemma)**.**

$$\mathcal{R}_\mathcal{S} \leq \max_{P\in\mathcal{P}} \|v_P\| \frac{\sqrt{2\ln|V_\mathcal{S}|}}{n} = \max_{P\in\mathcal{P}} \sqrt{\frac{2 f_\mathcal{S}(P)\ln|V_\mathcal{S}|}{n}}.$$

If we let $d = EVC((\mathcal{D}, R_\mathcal{P}), \mathcal{S})$ be the empirical VC-dimension on $\mathcal{S}$, then, using Sauer's Lemma [5], which links the size of a set to its VC dimension, we obtain that $|V_\mathcal{S}| \leq \sum_{i=0}^{d} \binom{n}{i} \leq (n+1)^d$. This then yields $\ln|V_\mathcal{S}| \leq d\ln(n+1)$, recovering Lemma 1.

In fact, [22] formulates a stronger version of the theorem as follows, which allows us to avoid using the empirical VC dimension of the sample, and to obtain tighter bounds.

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

$$w(s) = \frac{1}{s}\ln\left(\sum_{v\in V_\mathcal{S}} \exp\left(s^2\|v\|^2/(2n^2)\right)\right).$$

Then $\mathcal{R}_\mathcal{S} \leq \min_{s\in\mathbb{R}^+} w(s)$.

Computing w is infeasible as it would involve computing all supports. We then devise a function $\tilde{w}$, computable with a single scan of the sample, that upper bounds w and can hence be used to bound the Rademacher average.

Observation 1. *If we restrict the pattern set to the set $\mathcal{P}^+$ of patterns with strictly positive frequency, the maximum deviation of the frequency of patterns in $\mathcal{P}^+$ is the same as the one of patterns in $\mathcal{P}$. Hence, we can work with the set of all connected induced subgraphs of all the transactions, which is a huge but finite set, rather than the infinite set of all possible connected graphs.*

In the following paragraphs we show how to efficiently compute an upper bound to the Rademacher average considering the two scenarios of labelled graphs and unlabelled graphs.

Labeled graphs: In this paragraph we show how to efficiently bound $\mathcal{R}_\mathcal{S}$ for datasets of node-labeled graphs.

Note that we can see the individual labeled nodes as the building blocks of the transactions. Let L be the set of such labeled nodes. It is useful to consider them as patterns, as it will help to reuse the techniques used with labeled graphs when dealing with unlabeled ones.

We cover $V_\mathcal{S}$ as follows. Let $C_i = \{P = (V_P, E_P) \in \mathcal{P}^+$ s.t. $|V_P| = i\}$ be the set of patterns with i nodes and $V_i = \{v_P\ :\ P \in C_i\}$. Then $\bigcup_i V_i = V_\mathcal{S}$.

Then, consider the following partitioning of the sets V_i, $i \geq 1$. Assume to sort the labels according to some ordering, e.g. in increasing order by the frequency of their corresponding pattern in $\mathcal{S}$, and let the resulting order be $<_\ell$. Let $v_P \in V_i$ and let $Q \in L$ be s.t. Q is the first pattern in the ordering $<_\ell$ such that $Q \subseteq P$. Note that there always exists one such pattern, since transactions are non-empty. Then we assign v_P to the set $V_{i,Q}$.

Let $T_\mathcal{S}(i, Q)$ be the set of transactions $G = (V_G, E_G) \in \mathcal{S}$ s.t. $Q \subseteq G$ and at least i nodes have label $\geq_\ell Q$. Consider then a transaction $G = (V_G, E_G) \in T_\mathcal{S}(i, Q)$. Let $M_{i,Q,G}$ be the set of all induced connected subgraphs H of G s.t. H has i vertices, it contains pattern Q, and it does not contain patterns $Q' <_\ell Q$. Let also $m_{i,Q,G} = |M_{i,Q,G}|$. We then have the following lemma.

Lemma 3. *We have $|V_{i,Q}| \leq \sum_{G\in T_\mathcal{S}(i,Q)} m_{i,Q,G}$.*

Proof. We show that there exists an injective function $f : V_{i,Q} \longrightarrow \bigcup_{G\in T_\mathcal{S}(i,Q)} M_{i,Q,G}$. Let $v_P \in V_{i,Q}$, and first of all note that P cannot contain nodes $Q' <_\ell Q$, as v_P would belong to $V_{i,Q'}$. Then the pattern P, since it has positive frequency, is isomorphic to a connected induced subgraph H of some $G \in \mathcal{S}$, with G containing at least a node Q and at least i nodes with label $\geq_\ell Q$. Then $H \in M_{i,Q,G},\ G \in T_\mathcal{S}(i, Q)$. Let then $f(v_P) = H$. In case of multiple possible H's, ties are broken arbitrarily. Since P and $f(v_P)$ are isomorphic, no two different patterns can be mapped to the same connected induced subgraph H, and f is hence injective. Recalling that $|\bigcup_{G\in T_\mathcal{S}(i,Q)} M_{i,Q,G}| \leq \sum_{G\in T_\mathcal{S}(i,Q)} m_{i,Q,G}$, we have the claim. □

A slack bound to $m_{i,Q,G}$ is $\binom{|V_G|}{i}$. A better bound can be obtained noticing that nodes $Q' <_\ell Q$ cannot be chosen, and that at least one node Q must be chosen. Let $|V_G^{(\geq Q)}|$ be the number of nodes in G with label $\geq_\ell Q$ and $|V_G^{(>Q)}|$ be the number of nodes with label $>_\ell Q$. Then we have $m_{i,Q,G} \leq \binom{|V_G^{(\geq Q)}|}{i} - \binom{|V_G^{(>Q)}|}{i} \leq \binom{|V_G^{(\geq Q)}|}{i}$.

Note that in the bounds above we are not exploiting the fact that the subgraphs have to be connected. Then, if one is willing to sacrifice running times for obtaining better bounds, an exact subgraph enumeration algorithm can be suitably modified to provide a tighter bound to $m_{i,Q,G}$.

We now provide a second bound on $|V_{i,Q}|$, which exploits the fact that many patterns might share the same support.

Lemma 4. *We have $|V_{i,Q}| \leq 2^{|T_\mathcal{S}(i,Q)|} - 1$.*

Proof. Note that all transactions in $\mathcal{S} \setminus T_{\mathcal{S}}(i,Q)$ must have the corresponding entry in v_P set to 0 for all patterns $P \in V_{i,Q}$. Indeed, if a transaction has less than i nodes with label $\geq_\ell Q$ it cannot contain P, as it has i nodes and does not contain any node with label $<_\ell Q$. Moreover, since Q is an induced subgraph of P it is impossible for a transaction to contain P but not Q. Then there are only $2^{|T_{\mathcal{S}}(i,Q)|}$ possible assignments, one of which is of all zeros. □

Using the above we prove the following.

Lemma 5. *Define* $m_{i,Q} = \sum_{G \in T_{\mathcal{S}}(i,Q)} m_{i,Q,G}$ *and* $m'_{i,Q} = 2^{|T_{\mathcal{S}}(i,Q)|} - 1$. *Let also* χ *be the maximum number of nodes in any transaction. Let then* $\tilde{w} : \mathbb{R}^+ \to \mathbb{R}^+$ *be the function*

$$\tilde{w}(s) = \frac{1}{s} \ln \left(\sum_{i=1}^{\chi} \sum_{Q \in L} \min\{m_{i,Q}, m'_{i,Q}\} e^{\frac{s^2 |T_{\mathcal{S}}(i,Q)|}{2n^2}} \right)$$

Then we have that $w(s) \leq \tilde{w}(s)$, $\forall s \in \mathbb{R}$.

Proof. We can write $w(s)$ as follows.

$$w(s) \leq \frac{1}{s} \ln \left(\sum_{v_P \in V_{\mathcal{S}}} \exp\left(s^2 \|v_P\|^2 / (2n^2) \right) \right)$$
$$= \frac{1}{s} \ln \left(\sum_{i=1}^{\chi} \sum_{Q \in L} \sum_{v \in V_{i,Q}} e^{\frac{s^2 \|v\|^2}{2n^2}} \right)$$

Moreover, as shown in Lemma 4, all transactions in $\mathcal{S} \setminus T_{\mathcal{S}}(i,Q)$ must have the corresponding entry in v_P set to 0 for all patterns $P \in V_{i,Q}$. Then for such patterns we have $\|v_P\|^2 \leq |T_{\mathcal{S}}(i,Q)|$. Then we have

$$\sum_{v \in V_{i,Q}} e^{\frac{s^2 \|v\|^2}{2n^2}} \leq e^{\frac{s^2 |T_{\mathcal{S}}(i,Q)|}{2n^2}} \cdot |V_{i,Q}|$$

Combining the bounds on $|V_{i,Q}|$ from Lemmas 3 and 4 we obtain the claim. □

Observation 2. *By design, the two quantities* $m_{i,Q}$ *and* $m'_{i,Q}$ *are useful in different transaction size ranges. Indeed, the former is more powerful for small node counts* i*, where the cardinality of* $T_{\mathcal{S}}(i,Q)$ *is very large and thus not useful, while the number of connected subgraphs of size* i *is relatively small. For large node counts* i*, instead, the number of connected subgraphs is huge, but the size of* $T_{\mathcal{S}}(i,Q)$ *is much smaller, as large transactions are hopefully rare.*

We can compute all the needed quantities in a single scan of the sample $\mathcal{S}$, provided that the order $<_\ell$ is already available. Indeed, for each $G = (V_G, E_G) \in \mathcal{S}$, we check for the presence of patterns $P \in L$ in G. Then, for each $Q \in L$ in increasing order by $<_\ell$, if $Q \subseteq G$, we upper bound the quantity $m_{i,Q,G}$, either using the binomial formula or a subgraph enumeration algorithm. Moreover, we update the size of $T_{\mathcal{S}}(i,Q)$ for each $i = 1, \ldots, |V_G^{(\geq_\ell Q)}|$. Note that, if one is interested in patterns up to size k, it is sufficient to set $\chi = k$.

Then, we can minimize $\tilde{w}(s)$ using a nonlinear optimization solver to obtain a bound to $\mathcal{R}_{\mathcal{S}}$. The pseudocode of the procedure is provided in Algorithm 2.

Algorithm 2: ESTIMATERS($\mathcal{S}$)

```
m_{i,Q} = 0 ∀i, ∀Q ∈ L
for G ∈ S do
    for Q ∈ L do
        if Q ⊆ G then
            m_{i,Q} += (|V_G^{(≥Q)}| choose i) − (|V_G^{(>Q)}| choose i)
            for i = 1, ..., |V_G^{(≥Q)}| do
                T_{i,Q} ++
m'_{i,Q} = 2^{T_{i,Q}} − 1, ∀i, ∀Q ∈ L
w̃(s) = (1/s) ln( Σ_{i=1}^{χ} Σ_{Q∈L} min{m_{i,Q}, m'_{i,Q}} e^{s² T_{i,Q} / 2n²} )
return min_{s∈R+} w̃(s)
```

Unlabeled graphs: The method described above can be used for unlabeled graphs, as we can treat all nodes as having the same (empty) label. However this results in a fairly weak bound, as we are only subdividing $V_{\mathcal{S}}$ on the number of nodes of the patterns. Thus, rather than considering the single nodes as the building blocks of the transactions, we can consider some small subgraphs of size $h = 4$ or 5 instead. Then, we will compute exactly the frequency for all patterns of size up to h, and use the following covering for the sets V_i, $i > h$. Assume again to sort patterns of C_h according to some ordering $<_h$ (e.g. by frequency). Let $v_P \in V_i$ and let $Q \in C_h$ be s.t. Q is the first pattern in the ordering $<_h$ such that $Q \subseteq P$. Then we assign v_P to the set $V_{i,Q}$.

Note that, since in the labeled case we treated individual nodes as patterns, most of the analysis carries over to this case trivially, with the exception of the bounds to $m_{i,Q,G}$.

Indeed, while it is still true that $m_{i,Q,G} \leq \binom{|V_G|}{i}$, it is not possible to exploit the fact that patterns $Q' <_h Q$ should not be chosen to obtain a closed-form bound. Similarly, a subgraph enumeration procedure should be adapted as well.

We then have the following lemma, whose proof is akin to the one of Lemma 5:

Lemma 6. *Let* $m_{i,Q}$, $m'_{i,Q}$ *and* χ *be defined as before. Let then* $\tilde{w} : \mathbb{R}^+ \to \mathbb{R}^+$ *be the function*

$$\tilde{w}(s) = \frac{1}{s} \ln \Bigg(\sum_{i=1}^{h} \sum_{P \in C_i} e^{\frac{s^2 |T_{\mathcal{S}}(Q)|}{2n^2}} + \sum_{i=h+1}^{\chi} \sum_{Q \in C_k} \min\{m_{i,Q}, m'_{i,Q}\} e^{\frac{s^2 |T_{\mathcal{S}}(i,Q)|}{2n^2}} \Bigg).$$

Then we have that $w(s) \leq \tilde{w}(s)$, $\forall s \in \mathbb{R}$.

V. APPLICATIONS

This section describes two applications of the bounds on the VC-dimension and on Rademacher averages for subgraphs. In particular, Section V-A describes their use for approximate frequent subgraph mining through sampling, while Section V-B describes their use to solve the problem of true frequent subgraph mining.

A. Approximate Frequent Subgraph Mining with Guarantees

We now propose two algorithms, one based on the VC-dimension and one based on Radamacher Averages, to compute ε-approximations of frequent subgraphs by sampling. As with all sampling approaches, the main challenge is to bound the number of samples required to have guarantees on the relation between the results on the sample and the results on the whole dataset. In particular, we are interested in obtaining ε-close approximations.

A straightforward application of Chernoff bound yields that with a sample $\mathcal{D}'$ of $\frac{3}{\varepsilon^2}\ln\frac{2}{\delta}$ transactions from $\mathcal{D}$, for a *fixed* pattern P, with probability $1-\delta$, we have that $|f_{\mathcal{D}'}(P) - f_{\mathcal{D}}(P)| \leq \varepsilon$. One generally would then proceed to bound the probability of not obtaining a ε-close approximations using union bound over all patterns. In the case of frequent subgraph mining though this strategy fails due to the enormous number of patterns in $\mathcal{P}$, which is exponential in the square of the maximum number of nodes in a pattern.

We now present our algorithm based on the VC-dimension. In particular, the algorithm first computes the upper bound to the VC-dimension using Algorithm 1. Then, it takes a random sample $\mathcal{D}'$ of $\mathcal{D}$, taken uniformly at random, and reports in output $FG(\mathcal{D}', \mathcal{P}, \theta - \varepsilon/2)$, that is, all subgraphs with frequency $\geq \theta - \varepsilon/2$ in $\mathcal{D}'$. The following theorem (whose proof is analogous to the one in [21] for frequent itemsets mining) bounds on the number of samples that $\mathcal{D}'$ must contain for the output $FG(\mathcal{D}', \mathcal{P}, \theta - \varepsilon/2)$ to be a ε-approximation of $FG(\mathcal{D}, \mathcal{P}, \theta)$ with probability at least $1-\delta$.

Theorem 8. *Let $\mathcal{D}$ be a dataset of transactions (i.e., graphs), $\mathcal{P}$ a set of patterns (i.e., subgraphs), and v an upper bound to the VC dimension of the associated range space. Let $0 < \varepsilon, \delta < 1$. Let $\mathcal{D}'$ be a random sample of $\mathcal{D}$ (taken uniformly at random with replacements) of size*

$$\min\left\{|\mathcal{D}|,\ \frac{4c}{\varepsilon^2}\left(v + \ln\frac{1}{\delta}\right)\right\}$$

for some absolute constant c. Then $FG(\mathcal{D}', \mathcal{P}, \theta - \varepsilon/2)$ is an absolute ε-close approximation to $FG(\mathcal{D}, \mathcal{P}, \theta)$ with probability at least $1-\delta$.

As a second approach for obtaining ε-close approximations with guarantees is to use a progressive sampling scheme [22], with Rademacher averages to define stopping conditions. This avoids to process the entire dataset, which can be beneficial in extremely massive datasets, allowing to consider only a small sample of it. The outline of the sampling algorithm is the following:

1) at iteration i, obtain the random sample $\mathcal{D}_i$ from $\mathcal{D}$;
2) compute ε_R such that $\max_{P\in\mathcal{P}} |f_{\mathcal{D}'}(P) - f_{\mathcal{D}}(P)| \leq \varepsilon_R$;
3) check if $\varepsilon_R \leq \frac{\varepsilon}{2}$;
4) if so, return $\mathcal{D}' = \mathcal{D}_i$, else compute the sample size at iteration $i+1$, increase i and return to (1).

The computation of ε_R is performed using Rademacher averages. In particular, Lemma 5 (or Lemma 6, if the graphs are unlabelled) is used to compute an upper bound $\tilde{w}(s)$ to $w(s)$ for each $s \in \mathbb{R}$, which leads to an upper bound to the Rademacher average $\mathcal{R}_{\mathcal{S}}$ using Theorem 7. Such upper bound to $\mathcal{R}_{\mathcal{S}}$ is used to compute the probabilistic upper bound ε_R to $\max_{P\in\mathcal{P}} |f_{\mathcal{D}'}(P) - f_{\mathcal{D}}(P)|$ according to Theorem 2, which holds with probability at least $1-\delta$. Then, using the arguments employed in Theorem 8, we can prove that $FG(\mathcal{D}', \mathcal{P}, \theta - \varepsilon/2)$ is a ε-close approximation for $FG(\mathcal{D}, \mathcal{P}, \theta)$.

The last component in the progressive sampling algorithm is the choice of a sampling schedule, which is defined by the initial sample size and by the growth rate of the number of samples. Analogously to what is proved in [22], if the number of samples is smaller than $\frac{8\ln(2/\delta)}{\varepsilon^2}$, then the condition of Theorem 2 cannot be satisfied. So we choose $S_0 = \frac{8\ln(2/\delta)}{\varepsilon^2}$ as the initial sample size. As for the growth rate, one possibility would be to choose a geometric sampling schedule $|S_{i+1}| = \alpha|S_i|$, with α a parameter to be chosen by the user. In fact, as discussed in [22], a better sampling strategy is to use the estimated maximum deviation at iteration i, $\varepsilon_R = \min_s \tilde{w}(s) + \sqrt{\frac{2\ln 2/\delta}{|S_i|}}$, to generate the next sample size. Indeed, since there is a quadratic dependency between the sample size and the maximum deviation, a good guess for the next sample size is $|S_{i+1}| = \left(\frac{2\varepsilon_R}{\varepsilon}\right)^2 |S_i|$.

B. True Frequent Subgraph Mining

As described in the introduction, a common scenario [24] in frequent pattern mining is that the dataset $\mathcal{D}$ is not to be considered as the ground truth on the underlying generating process, but rather as a sample from an unknown generating distribution π. In this setting then one would like to extrapolate, from the sample $\mathcal{D}$, the patterns that are frequent in the underlying distribution. More formally, the frequent patterns are all the patterns P such that $\sum_{G:P\subseteq G} p_\pi(G) \geq \theta$, where p_π is the density function of π.

In this context we have to define a more general notion of ε-approximation.

Definition 9. *Let (X, R) be a range space and let π a distribution on X. For $0 < \varepsilon < 1$, a bag B of elements of X is an ε-approximation for (X, π) if, for all $r \in R$ we have*

$$\left| p_\pi(r) - \frac{|B \cap r|}{|B|} \right| \leq \varepsilon$$

where $p_\pi(r) = \sum_{x\in r} p_\pi(x)$.

A simple algorithm to compute a ε-approximation of the set of true frequent subgraphs with probability at least $1-\delta$ is analogous to the one proposed in [24] for true frequent itemset mining, and works as follows: using dataset $\mathcal{D}$, compute an upper bound $\varepsilon/2$ to $\max_{P\in\mathcal{P}} |p_\pi(P) - f_{\mathcal{D}}(P)|$ that holds with probability at least $1-\delta$; report in output the set of patterns $FG(\mathcal{D}, \mathcal{P}, \theta - \varepsilon/2)$ with frequency at least $\theta - \varepsilon/2$ in the dataset $\mathcal{D}$. The output is then a ε-approximation of the set of true frequent subgraphs with probability at least $1-\delta$ (the proof is analogous to the one in [24]).

The upper bound $\varepsilon/2$ to $\max_{P\in\mathcal{P}} |p_\pi(P) - f_{\mathcal{D}}(P)|$ that holds with probability at least $1-\delta$ can be computed using either the VC-dimension (see Section III) or Rademacher

TABLE I: Key properties of datasets. See Section II-A for the definition of the properties.

Name	$\lvert\mathcal{D}\rvert$	$\lvert L_V\rvert$	$\lvert L_E\rvert$	avg. $\lvert V\rvert$	avg. $\lvert E\rvert$
Akos	10000000	76	2	49.6	51.6
Pubchem	25000000	103	2	50.1	52.1
Reddit	203088	1	1	23.9	24.9
Alphafold	541374	20	3	353.8	373.8

averages (see Section IV). For VC-dimension, while the results of Theorem 1 hold even for this general definition of ε-approximations, albeit with the caveat that sampling has to be performed according to π rather than uniformly, it is impossible to bound the VC dimension of the range space $(\Pi, \mathcal{P})$ associated with π directly, as the support Π of π is in general unknown. One has then to resort to the weaker bound based on the empirical VC dimension on the sample $\mathcal{D}$ (provided by Lemma 1).

The VC dimension of $(\mathcal{D}, R_{\mathcal{P}})$ can be bounded using the c-bound, as described in the previous sections, giving then a bound on the empirical VC dimension of $(\Pi, R_{\mathcal{P}})$ on $\mathcal{D}$. This immediately yields, by Lemma 1, a bound on the maximum deviation of the frequencies of patterns in $\mathcal{D}$ with respect to their true frequencies in the generating distribution π.

Rademacher-averages instead provide bounds requiring only *sample dependent* quantities, which can, thus, be directly used to estimate the maximum deviation of frequencies even when the dataset $\mathcal{D}$ is to be considered as a sample from an unknown generating distribution. That is, one can obtain the upper bound $\varepsilon/2$ to $\max_{P\in\mathcal{P}} |p_\pi(P) - f_{\mathcal{D}}(P)|$ as described in Section IV considering the whole dataset $\mathcal{D}$ as the sample (denoted with $\mathcal{S}$ in Section IV).

VI. Experiments

We performed an extensive experimental evaluation on real datasets to assess our bounds and methods. In particular, the main goals of the experimental evaluation are: to compare the tightness of the bounds we presented both in the *sampling from a static dataset* and in the *sampling from a distribution* version of the problem; when sampling from a fixed dataset, assess the performance benefits in terms of running time and peak memory consumption; assess the precision, recall and maximum deviation of our mining algorithms.

We implemented our algorithms in C++ and used no form of parallelism. The code is available at [1]. All experiments were performed on a server with a dual Intel Xeon 5220 processor with 72 cores and 1Tb of RAM. Moreover, in order to perform the subgraph isomorphism checks within our methods, we have implemented a wrapper around the VF3Lib library [6], as it is reportedly one of the fastest ones. As the nonlinear optimizer, we used the NLopt library [2]. For what concerns the subgraph mining step, we used the Gaston library [16]. This mining algorithm, as shown in [33] and more recently in [32], is currently the state-of-the-art in subgraph mining. Nonetheless, we remark that our approximation algorithm is completely orthogonal to improvements in the mining step, and in our mining pipeline Gaston can be easily replaced by (faster) miners that could be developed by future works. Gaston has two versions, one with occurrence lists, which is fast but uses a significant amount of memory, and one without occurrence lists (Gaston RE) that uses a small amount of memory but is significantly slower than the other version. For labelled datasets, where the search space can be efficiently pruned, we were able to use the former version, while for unlabelled datasets the mining process on Gaston with occurrence lists exceeded one terabyte of main memory even for small subgraph sizes and high-frequency thresholds, so we had to resort to Gaston RE.

We used datasets from various fields, including computational chemistry, computational biology and social network analysis. The AKOS and PUBCHEM datasets are subsets of two well-known molecular databases[1], converted from SMILES to a suitable graph format, and have already been used for validating large-scale subgraph mining [13]. REDDIT[2] is a collection of graphs representing threads collected from Reddit in May 2018 and is the only unlabeled dataset. Moreover, we introduce a new graph dataset ALPHAFOLD based on the protein structure predictions of DeepMind's Alphafold on the SwissProt dataset[3]. We generated the graphs, using the Graphein library, using the amino acids as nodes and peptide, hydrogen and aromatic bonds as edges. Since many proteins feature a high node count, this dataset can be considered a stress test for our methods, as their performance should degrade as the average size of the graphs grows. Table I shows some of the key properties of the datasets.

A. Approximate Frequent Subgraph Mining

In this section we explore the performance of our sampling algorithms for approximate frequent subgraph mining (see Section V-A), with the goal of obtaining accurate approximations while improving running times and memory usage.

We first compared the bound on sample size obtained using the c-bound and VC-dimension for various maximum subgraph sizes k and the bound obtained using the progressive sampling approach with Rademacher averages. We fixed $\delta = 0.05$ (we also tested other values, but, since the sample size has only a logarithmic dependency from δ, the results are almost identical), and let ε range in $\{0.02, 0.04, 0.1, 0.2\}$. The comparison of the sample sizes is shown in Fig. 5, which reports the mean values over 5 runs, together with 95% confidence interval (in shaded color). For all datasets, the best bound provides a sample size that is less than 10% of the whole dataset already for $\varepsilon = 0.1$. In all the datasets the VC-dimension bounds significantly outperform the bounds based on Rademacher averages. This might be due to some slackness in the bound provided by Massart leamma or due to the progressive sampling approach employed by the Rademacher-averages-based algorithm, which may lead to conservative (i.e., larger) estimates of the required sample sizes.

[1] http://akosgmbh.de/, https://pubchem.ncbi.nlm.nih.gov
[2] https://snap.stanford.edu/data
[3] https://alphafold.ebi.ac.uk/

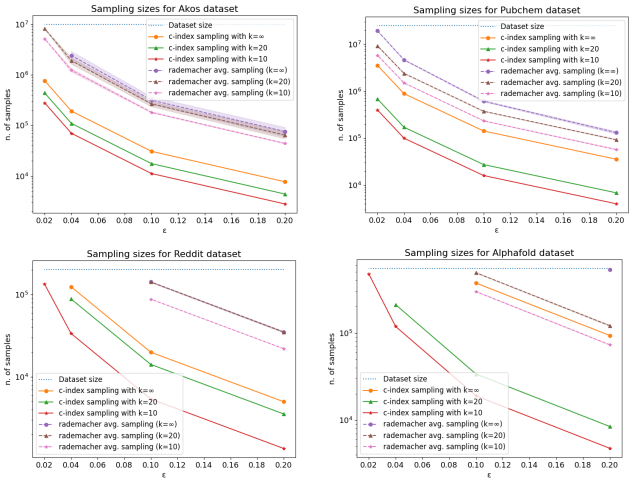

Fig. 5: Sample size bounds using the VC dimension bound based on the c-bound and using Rademache averages. Values are mean over 5 runs. 95% confidence interval are shaded.

TABLE II: Running time and peak memory usage to obtain the sample for approximate frequent subgraph mining ($\varepsilon = 0.1$).

	VC-dimension		Rademacher	
Dataset	Time (s)	Mem. (MB)	Time (s)	Mem. (MB)
Akos	108 ± 2	23 ± 1	113 ± 13	36 ± 1
Pubchem	270 ± 1	102 ± 0	251 ± 3	47 ± 0
Reddit	1.2 ± 0.1	21 ± 0	34.6 ± 0.8	231 ± 1
Alphafold	95 ± 2	643 ± 0	1900 ± 11	55 ± 0

Interestingly, as the maximum subgraph size in the c-bound bound decreases, the sample size decreases as well. Thanks to this property, when one wants to limit the mining process to small subgraphs, one can obtain significantly smaller sample sizes. The same holds, albeit to a more limited extent, for the Rademacher averages bounds.

We then compared the running times and peak memory usage to obtain the sample (i.e., compute the sample size and perform the sampling) using our approaches, for all datasets and $\varepsilon = 0.1$. We run the sampling methods 5 times and report the mean of the quantities, together with the standard deviation. As shown in Table II, the c-bound-based method is faster. This may, again, be due to the progressive sampling approach required by the Rademacher averages method. Moreover, as expected, computing the c-bound becomes faster as k decreases, as the running times depend on the actual c-bound, and does not depend on ε, as the entire dataset has to be considered anyways (results omitted for space constraints). On the other hand, the Rademacher-based method performance grows quadratically with $1/\varepsilon$, as it depends on the number of samples to be considered.

We then evaluate the quality of the approximation obtained by our best method, that is the VC-dimension-based approach. We considered $\varepsilon = 0.02$ and $\varepsilon = 0.1$, using $k = 10$ for Alphafold and $k = 20$ for the other datasets, which are conservative bounds to the maximum size of a frequent subgraph, for the c-bound computation. We considered $\theta = 0.2, 0.5, 0.8$ for all datasets except ALPHAFOLD, where we used $\theta = 0.1, 0.2, 0.5$ as the number of frequent patterns at higher thresholds is particularly low. For Reddit, the exact mining approach (on the whole dataset) could not complete, therefore we could not assess the quality of our approximations. Moreover, as stated before, since Reddit is unlabelled, we had to use Gaston RE to mine it due to memory issues. (Note that the output of the two versions of the miner is exactly the same.) Since Gaston RE is quite slow, it completed only for the subsampled dataset corresponding to $\varepsilon = 0.1$ ($\varepsilon = 0.02$ would exceed several weeks of running time). Moreover, when mining this dataset we limited the subgraph sizes to 7, as already for 8 the computation needed several weeks.

TABLE III: Precision, recall and max. deviation of approximate solutions

		$\varepsilon = 0.02$			$\varepsilon = 0.1$		
Dataset		Prec.	Recall	Max dev.	Prec.	Recall	Max dev.
Akos	$\theta = 0.8$	0.99	1.0	0.0012	0.81	1.0	0.0046
	$\theta = 0.5$	0.95	1.0	0.0019	0.77	1.0	0.0112
	$\theta = 0.2$	0.89	1.0	0.0025	0.51	1.0	0.0112
Pubchem	$\theta = 0.8$	0.85	1.0	0.0014	0.71	1.0	0.0050
	$\theta = 0.5$	0.96	1.0	0.0015	0.78	1.0	0.0113
	$\theta = 0.2$	0.90	1.0	0.0015	0.52	1.0	0.0113
Alphafold	$\theta = 0.5$	0.99	1.0	0.007	0.93	1.0	0.0084
	$\theta = 0.2$	0.90	1.0	0.0008	0.56	1.0	0.0091
	$\theta = 0.1$	0.87	1.0	0.0008	0.47	1.0	0.0091

Table III shows the recall, i.e. the ratio of the number of true positives and the number of actually frequent patterns, the precision, i.e. the ratio of the number of true positives and the number of returned patterns, and the maximum deviation of the reported frequency of the true positives. The reported precisions and recalls are averages over 5 runs, and since the standard deviation is always below 0.01, we omit it. The reported maximum deviations are the maximum over the 5 runs. Remarkably, the number of false negatives, throughout all of our tests, is always zero, and hence our approximate mining method has always recall 1.0. This supports the fact that the probability of reporting at least a false negative is less than δ, as stated by Theorem 8. In fact, in our experiments the algorithm never yielded false negatives, suggesting that in practice the probability of losing frequent subgraphs is even lower than this bound. For what concerns the precision, we point out that our methods have no control over it, as it depends on the number of subgraphs with frequency in $[\theta - \varepsilon/2, \theta]$, which is unknown a priori. Nonetheless, the experiments show that, especially for $\varepsilon = 0.02$, the number of false positives is quite limited. Moreover, all false positives are "acceptable" ones, in the sense that their frequencies are never more than ε lower than the threshold, as required by Definition 2. Finally, the maximum deviation of the frequency of truly frequent patterns is noticeably smaller than the guarantee provided by the theory, and the errors are quite concentrated around 0 (see Fig. 6), especially for high thresholds. This hints that tighter bounds might be possible to obtain. We remark that for computing the above quantities, we first had to solve the

TABLE IV: Performance of exact and approximate frequent subgraph mining algorithms

Dataset		Full dataset			$\varepsilon = 0.02$			$\varepsilon = 0.1$		
		Samples	Time (s)	Mem. (GB)	Samples	Time (s)	Mem. (GB)	Samples	Time (s)	Mem. (GB)
Akos	$\theta = 0.8$	$10 \cdot 10^6$	690 ± 2	93 ± 0	439979	30.8 ± 0.4	4.3 ± 0	17599	1.1 ± 0.1	0.1 ± 0
	$\theta = 0.5$		2030 ± 297	105 ± 0		82 ± 2	4.8 ± 0		2.0 ± 0.1	0.2 ± 0
	$\theta = 0.2$		9160 ± 534	118 ± 0		462 ± 45	5.5 ± 0		12.5 ± 1.1	0.2 ± 0
Pubchem	$\theta = 0.8$	$25 \cdot 10^6$	1400 ± 10	227 ± 0	684979	38 ± 0.4	6.3 ± 0	27400	1.4 ± 0	0.2 ± 0
	$\theta = 0.5$		4303 ± 454	251 ± 0		105 ± 8	6.9 ± 0		3.8 ± 0.3	0.3 ± 0
	$\theta = 0.2$		14415 ± 713	279 ± 0		395 ± 25	7.7 ± 0		13.6 ± 0.9	0.3 ± 0
Reddit	$\theta = 0.8$	203088	-	-	69979	-	-	2800	32867 ± 3293	0.01 ± 0
	$\theta = 0.5$		-	-		-	-		46357 ± 11170	0.01 ± 0
	$\theta = 0.2$		-	-		-	-		81780 ± 18619	0.01 ± 0
Alphafold	$\theta = 0.5$	541374	157 ± 4	9.3 ± 0	474979	144 ± 14	8.2 ± 0	19000	4.6 ± 0.2	0.3 ± 0
	$\theta = 0.2$		310 ± 56	9.5 ± 0		302 ± 36	8.3 ± 0		9.2 ± 1.3	0.3 ± 0
	$\theta = 0.1$		542 ± 78	9.7 ± 0		436 ± 28	8.5 ± 0		14.8 ± 3.1	0.3 ± 0

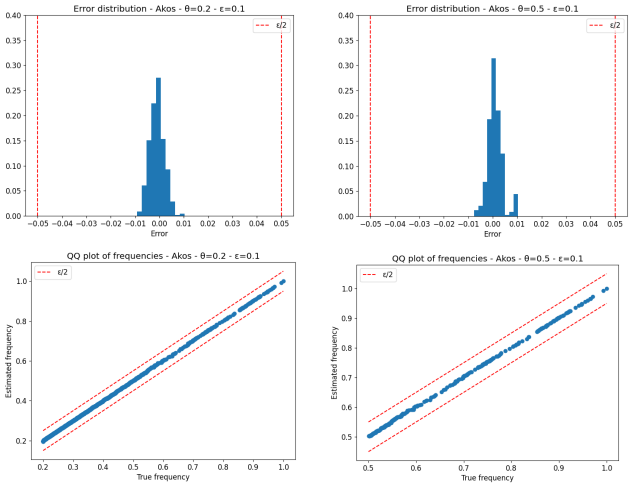

Fig. 6: Deviation distribution of estimated frequencies for approximate subgraph mining. We use the VC dimension bound approach with $k = 10$ for Alphafold and $k = 20$ for the other datasets.

subgraph mining problem on the original full dataset, which is an expensive computation that in some cases (e.g. on the REDDIT dataset) is basically impossible to carry out. In these latter cases the theoretical bounds we provided are the *only guarantees* that one might have on the quality of the solution on the sample.

Lastly, Table IV shows the performance gains in executing the mining algorithm on the sampled datasets in terms of running times and peak memory consumption, again using the VC-dimension-based approach. The performance of the mining algorithm not only depends on the number of transactions in the dataset, but also on the number of frequent patterns. Since in the sampled datasets we have to mine all patterns with frequency $\theta - \varepsilon/2$, the number of outputted patterns will be higher than the number of true frequent patterns, and this could worsen running times and memory consumption. Note that the performance benefits are the primary reason to use the sampled datasets, hence we hope that the reduction in the input size overcomes the increase in the output size in terms of impact on the performance, and to hence see substantial improvements. Indeed, as reported in Table IV, we see up to a 20x reduction in peak memory usage and running times for $\varepsilon = 0.02$ and up to a 1000x reduction for $\varepsilon = 0.1$. This makes it possible to run the subgraph mining procedure on a large dataset such as PUBCHEM on a standard laptop with 8GB of RAM rather than on a dedicated system with hundreds of gigabytes of RAM. Moreover, for challenging datasets such as REDDIT, the sampling scheme allows to conclude the mining procedure in a few days of computation rather than in weeks or months, making the problem from virtually impossible to solve to solvable. We also assessed how the total computing time is subdivided among computing the sample size, sampling the dataset, and running the mining procedure on the sample (see Fig. 7). We note that, especially for low values of ε, the time needed to compute the bound and to sample the dataset is a small portion of the time needed to mine the frequent patterns.

B. True Frequent Subgraph Mining

We assessed our approaches for the true frequent subgraph mining task (see Section V-B). To obtain datasets with a known ground truth, we use the datasets described in Table I as the generating distribution, and sample them uniformly and with replacement. Note that in this way the probability that a subgraph appears in a transaction from the underlying distribution is equal to its frequency in the original dataset. We generated, for each dataset, a set of samples of sizes in the range $[10^4, 10^6]$.

We first compared the bounds on the maximum deviation $\varepsilon/2$ obtained using the empirical VC-dimension (EVC) bound and Rademacher averages bound. Fig. 8 reports averages over 5 runs together with 95% confidence intervals. We note that for this task the Rademacher averages bound significantly outperforms EVC ones. Moreover, limiting the maximum pattern size leads to sharper bounds.

We then assessed the quality of the approximation provided by our best performing method for true frequent subgraph mining, i.e. the Rademacher averages one, limiting the maximum pattern size to $k = 10$ for Alphafold and $k = 20$ for the other datasets. Table V shows precision, recall and maximum deviation of the approximate frequencies. (Note

TABLE V: Precision, recall and maximum deviation for true frequent subgraph mining

Dataset		$n = 10^4$				$n = 10^5$				$n = 10^6$			
		Bound	Precis.	Recall	Max dev.	Bound	Precis.	Recall	Max dev.	Bound	Precis.	Recall	Max dev.
Akos	$\theta = 0.8$	0.232	0.39	1.0	0.0064	0.075	0.62	1.0	0.0018	0.026	0.93	1.0	0.0008
	$\theta = 0.5$		0.15	1.0	0.0124		0.62	1.0	0.0038		0.84	1.0	0.0015
	$\theta = 0.2$		-	-	-		0.32	1.0	0.0040		0.72	1.0	0.0015
Pubchem	$\theta = 0.8$	0.285	0.24	1.0	0.0051	0.091	0.58	1.0	0.0032	0.029	0.79	1.0	0.0012
	$\theta = 0.5$		0.13	1.0	0.0165		0.69	1.0	0.0043		0.89	1.0	0.0014
	$\theta = 0.2$		-	-	-		0.26	1.0	0.0053		0.70	1.0	0.0016
Alphafold	$\theta = 0.5$	0.260	0.48	1.0	0.0077	0.083	0.91	1.0	0.0041	0.026	0.97	1.0	0.0010
	$\theta = 0.2$		-	-	-		0.36	1.0	0.0041		0.74	1.0	0.0017
	$\theta = 0.1$		-	-	-		-	-	-		0.19	1.0	0.0017

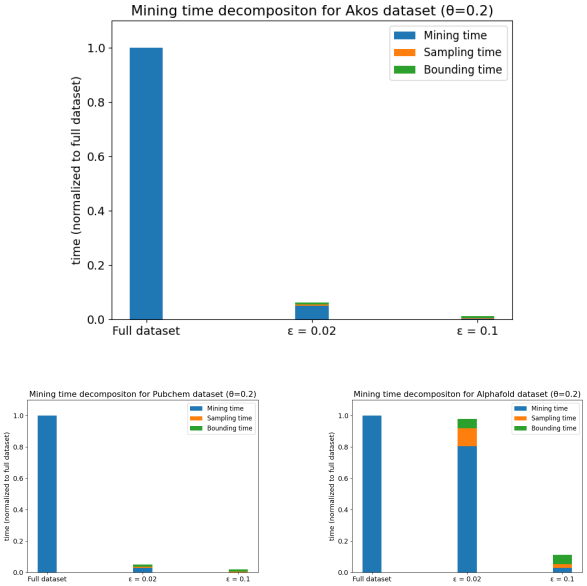

Fig. 7: Running time decomposition for approximate frequent subgraph mining. We use the VC dimension bound approach with $k = 10$ for Alphafold and $k = 20$ for the other datasets.

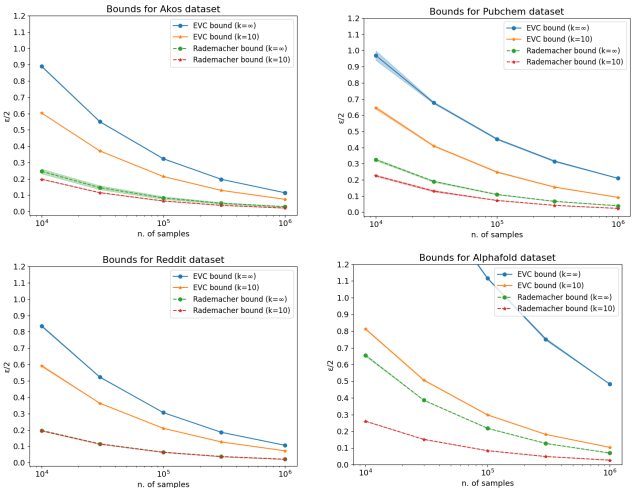

Fig. 8: Maximum deviation bounds using the empirical VC-dimension bound and using Rademacher averages. Values are means over 5 runs. 95% confidence intervals are shaded.

that such quantities can be computed since we know the underlying generative distribution of the datasets, as explained at the beginning of this section). As expected, as the bound on the maximum deviation decreases with the increasing sample size, the precision of the approximation increases for all datasets and thresholds. Moreover, as for approximate frequent subgraph mining, the recall of the method is always 1.0, that is, there are no false negatives. Moreover, the actual maximum deviations of the estimated frequencies from the true ones are much smaller that the bounds provided by Rademacher averages, suggesting that sharper bounds may be possible.

VII. Conclusions

In this work, we derived efficiently computable bounds on the VC-dimension and on the Rademacher averages of subgraphs. We showed that bounds can be used to obtain efficient approximations for two graph mining tasks: frequent subgraph mining and true frequent subgraph mining. Our extensive experimental evaluations shows that our bounds, and the corresponding algorithms, result in high-quality approximations for both applications on several on real datasets.

Acknowledgements

Part of this work was supported by the MIUR, the Italian Ministry of Education, University and Research, under PRIN Project n. 20174LF3T8 AHeAD (Efficient Algorithms for HArnessing Networked Data) and project National Centre for HPC, Big Data and Quantum Computing, CN00000013, and by the University of Padova under project SEED 2020 RATED-X.

References

[1] https://github.com/VandinLab/SubgraphMining_ICDE2023
[2] Nlopt. https://nlopt.readthedocs.io/.
[3] Alobaidi and Allami. The use of frequent subgraph mining to develop a recommender system for playing real-time strategy games. *ICDM* 2019.
[4] Borgelt and Berthold. Mining molecular fragments: Finding relevant substructures of molecules. *ICDM* 2002.
[5] Boucheron, Bousquet, and Lugosi. Theory of classification: a survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.

[6] Carletti, et al. Challenging the time complexity of exact subgraph isomorphism for huge and dense graphs with vf3. *IEEE Trans. Pattern Analysis and Mach. Intelligence* 2018.
[7] Deshpande et al. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Trans. on Knowledge and Data Engineering*, 2005.
[8] Fan et al. Association rules with graph patterns. *Proceedings of the VLDB Endowment*, 2015.
[9] Guralnik and Karypis. A scalable algorithm for clustering sequential data. *ICDM* 2001.
[10] Inokuchi, Washio, and Motoda. An apriori-based algorithm for mining frequent substructures from graph data. *PKDD* 2000.
[11] Koltchinskii and Panchenko. Rademacher processes and bounding the risk of function learning. In *High dimensional probability II*, 2000.
[12] Kuramochi and Karypis. Frequent subgraph discovery. *ICDM* 2001.
[13] Lin, Xiao, and Ghinita. Large-scale frequent subgraph mining in mapreduce. *ICDE* 2014.
[14] Löffler and Phillips. Shape Fitting on Point Sets with Probability Distributions. *ESA* 2009.
[15] Mrzic, et al. Grasping frequent subgraph mining for bioinformatics applications. *BioData mining*, 2018.
[16] Nijssen and Kok. The gaston tool for frequent subgraph mining. *Electron. Notes TCS*, 2005.
[17] Pellegrina et al. Mcrapper: Monte-carlo rademacher averages for poset families and approximate pattern mining. *KDD* 2020.
[18] Pellegrina and Vandin. Silvan: Estimating betweenness centralities with progressive sampling and non-uniform rademacher bounds. *arXiv:2106.03462*, 2021.
[19] Preti, De Francisci Morales, and Riondato. Maniacs: Approximate mining of frequent subgraph patterns through sampling. *KDD* 2021.
[20] Riondato and Kornaropoulos. Fast approximation of betweenness centrality through sampling. *Data Mining and Knowledge Discovery*, 2016.
[21] Riondato and Upfal. Efficient discovery of association rules and frequent itemsets through sampling with tight performance guarantees. *ACM Trans. KDD*, 2014.
[22] Riondato and Upfal. Mining frequent itemsets through progressive sampling with rademacher averages. *KDD* 2015.
[23] Riondato and Upfal. Abra: Approximating betweenness centrality in static and dynamic graphs with rademacher averages. *ACM Trans. KDD*, 2018.
[24] Riondato and Vandin. Finding the true frequent itemsets. *SDM* 2014.
[25] Riondato and Vandin. Misosoup: Mining interesting subgroups with sampling and pseudodimension. *ACM Trans. KDD*, 2020.
[26] Santoro, Tonon, and Vandin. Mining sequential patterns with vc-dimension and rademacher complexity. *Algorithms*, 2020.
[27] Servan-Schreiber, Riondato, and Zgraggen. Prosecco: Progressive sequence mining with convergence guarantees. *Know. and Inf. Sys.*, 2020.
[28] Seshadhri and Tirthapura. Scalable subgraph counting: the methods behind the madness. *WWW* 2019.
[29] Shalev-Shwartz and Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.
[30] Vapnik. *Statistical learning theory*. Wiley, New York, 1998.
[31] Wang et al. Scalable mining of large disk-based graph databases. *KDD* 2004.
[32] Welke. Efficient Frequent Subgraph Mining in Transactional Databases. *DSAA* 2020.
[33] Wörlei et al. A quantitative comparison of the subgraph miners mofa, gspan, ffsm, and gaston. *PKDD 2005*.
[34] Yan and Han. gspan: Graph-based substructure pattern mining. *ICDM* 2002.