Efficient Privacy-Preserving Recommendations based on Social Graphs

Conference Paper · June 2019

DOI: 10.1145/3298689.3347013

Aidmar Wainakh
Technische Universität Darmstadt

Tim Grube
Technische Universität Darmstadt

Jörg Daubert
Philipps University of Marburg

Max Mühlhäuser
Technische Universität Darmstadt

Efficient Privacy-preserving Recommendations based on Social Graphs

Aidmar Wainakh
wainakh@tk.tu-darmstadt.de
Technische Universität Darmstadt
Darmstadt, Germany

Tim Grube
grube@tk.tu-darmstadt.de
Technische Universität Darmstadt
Darmstadt, Germany

Jörg Daubert
daubert@mathematik.uni-marburg.de
Philipps-Universität Marburg
Marburg, Germany

Max Mühlhäuser
max@tk.tu-darmstadt.de
Technische Universität Darmstadt
Darmstadt, Germany

ABSTRACT

Recommender systems use association rules mining, a technique that captures relations between user interests and recommends new potential ones accordingly. Applying association rule mining causes privacy concerns as user interests may contain sensitive personal information (e.g., political views). This potentially even inhibits the user from providing information in the first place. Current distributed privacy-preserving association rules mining (PPARM) approaches use cryptographic primitives that come with high computational and communication costs, rendering PPARM unsuitable for large-scale applications such as social networks. We propose improvements in the efficiency and privacy of PPARM approaches by minimizing the required data. We propose and compare sampling strategies to sample the data based on social graphs in a privacy-preserving manner. The results on real-world datasets show that our sampling-based approach can achieve a high average precision score with as low as 50% sampling rate and, therefore, with a 50% reduction of communication cost.

CCS CONCEPTS

• **Security and privacy** → **Social network security and privacy**.

KEYWORDS

privacy-preserving data mining, distributed data, recommender system, association rules, efficiency

ACM Reference Format:
Aidmar Wainakh, Tim Grube, Jörg Daubert, and Max Mühlhäuser. 2019. Efficient Privacy-preserving Recommendations based on Social Graphs. In *Proceedings of RecSys '19: ACM Conference on Recommender Systems (RecSys '19).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

RecSys '19, September 16–20, 2019, Denmark

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Online social networks (OSNs) are becoming an essential means of communication in our modern society; people increasingly use the services provided by OSNs in their daily life. Many of these services are based on recommender systems which reveal interesting and "new" content to individuals, e.g., suggesting new songs in Spotify, or recommending "old" friends on Facebook. Today's recommender systems use data mining and machine learning methods to create the recommendations [25], one of the most efficient methods is association rules mining (ARM) [34]. ARM captures the relations between items; these relations lead from known interesting items of users to potentially interesting new items.

Currently, the dominant OSNs (e.g., Facebook and Spotify) realize recommender systems in a centralized fashion, i.e., the providers themselves collect and process the user data, and mine the association rules. Unfortunately, the providers show repeatedly insufficient commitment to the privacy of the users [22].The data of users are oftentimes used without informed consent or misused in different ways; regularly, the providers disclose data to third parties (e.g., data broker companies).Moreover, user data was prone to unauthorized access in several occasions (e.g., Facebook tokens hack 2018 [15]); some parties violated the usage policy of the OSNs and collected user data for suspicious purposes (e.g., Cambridge Analytica [14]). The privacy of users in centralized OSNs is regularly and seriously endangered or even violated considering the aforementioned issues.

To address this privacy concern, several approaches were proposed to build association rules in a privacy-preserving manner. These privacy-preserving association rules mining (PPARM) approaches can be classified into two categories: first, approaches that anonymize the data *prior* to the mining process [10, 13]. Second, approaches that use cryptography to *preserve the privacy* of the data during the mining process [6, 32]. In distributed systems, the cryptography-based approaches provide state-of-the-art solutions. However, these approaches are intended to be applied in distributed systems with a *limited* number of users. The number of necessary cryptographic operations impairs the efficiency of the PPARM considerably in today's systems with increasing complexity.

Improving the efficiency of PPARM is indispensable to improve the protection of user privacy. In this paper, we contribute a novel combination of graph sampling and distributed PPARM. In this combination, we achieve an improvement of efficiency by reducing the number of involved users by applying distributed graph sampling. The sampling process does not only promote efficiency, but it also enhances privacy as less data is used to derive the association rules. We exploit the similarity of connected users (see the theory of homophily [12, 23]), i.e., we utilize common interests of "friends", to represent local groups of users with only a few (maybe even only one) representatives. Our results show that the inherent correlation of user data and the social structure of the underlying graph can be used to improve the efficiency of the PPARM while maintaining a high accuracy of the derived rules.

The remainder of this paper is organized as follows. First, we present background on ARM and sampling under our problem setting in Section 2. Then, we present related work on PPARM and efficient ARM in Section 3. In Section 4, we introduce an overview of our approach, followed by a detailed description in Sections 5 and 6. Section 7 discusses our results. Finally, Section 8 summarizes our contributions, draws conclusions, and points out future work.

2 BACKGROUND AND PROBLEM SETTING

The contribution of this paper addresses sampling for ARM to work with less data (efficiency and data minimization) while preserving privacy. This section introduces (1) the basics of ARM and (2) sampling.

2.1 ARM and Problem Setting

ARM captures the relation of items by (a) identifying items that occur frequently together—frequent itemsets—and (b) establishing rules between these itemsets. In OSNs, items refer to a user's interest. An interest is expressed by the user posting/consuming content, e.g., writing or liking a post.

With common centralized OSNs, like Facebook and Twitter, the provider stores and controls all items of all users. In other words: the provider knows everything about every user. That leads to the following challenge having the users' desire to protect their privacy in mind: protect items from the provider and from other users. As complete protection (unobservability) would defy the very purpose of an OSN, we relax that requirement to privacy in terms of *unlinkability*: neither the provider nor other users should be able to link items of a user, i.e., determine itemsets for any user. That leads to the following problem statement for this paper:

i. Protect the user's itemsets from other users.
ii. Protect the user's itemsets from the provider.
iii. The provider has full knowledge about the social graph, which is the friendship connections between the users.
iv. The provider and users follow the semi-honest adversary model.

This problem statement is referred to as privacy-preserving association rules mining (PPARM). PPARM either operates on encrypted items in a central place or by distributing items, i.e., leaving the items and, therefore, sensitive itemsets, with their users. For the remainder of this paper, we focus on distributed PPARM and formalize the problem as follows:

We refer to a single piece of content as an *item* i from the set $I = \{i_1, i_2, \ldots, i_m\}$ of all possible items. Let the set $U = \{u_1, u_2, \ldots, u_n\}$ cover all users of a system. The *interested in* relation $r_{u,i}$ models a user's u interest in an item i. We denote this relation as

$$r_{u,i} = \begin{cases} 1 & \text{if } u \text{ is interested in } i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

For each user $u \in U$, let T be the set of interesting items for u such that $T \subseteq I$ and $\forall i \in T, r_{u,i} = 1$. A dataset D is then a set of the users' interesting items $D = \{T_1, T_2, ..., T_n\}$. ARM derives *frequent itemsets* from a dataset D. The *support* $s_D(X)$ of an itemset $X \subseteq I$ is the ratio of users that their items T in D contain X. An itemset X is frequent, if its support exceeds the *support threshold* θ, i.e., if $s_D(X) \geq \theta$ – the set FI_D summarizes the frequent itemsets found in the dataset D. Given two distinct itemsets X, Y with $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$, an *association rule* ar_i is an implication $X \longrightarrow Y$. For each association rule ar_i, the confidence $c_D(ar_i)$ indicates how often the rule is found to be true. Association rules ar_i that exceed the *confidence threshold* β, i.e., $c_D(ar_j) \geq \beta$, are found to be *reliable* in D. The problem space of ARM covers two main challenges:

i. Derivation of the itemsets FI_D under θ.
ii. Establishing *reliable* association rules of FI_D under β.

The main costs emerge from the challenge (1) as the whole dataset D needs to be analyzed [6]. In the focus of this paper, our contribution provides an efficiency improvement for the subproblem of deriving the frequent itemsets, while we also improve the users' privacy by reducing the number of involved users in this process.

2.2 Sampling

Sampling is the selection of a subset of a population to estimate characteristics of the entire population. The most relevant factors for sampling are (1) the sampling (selection) method and (2) the size of the sample. As the related work covers sampling sizes extensively, even for ARM [8, 27], this paper takes a closer look at sampling methods for ARM with privacy in mind.

As we propose using the social graph as a base for sampling, we consider different graph sampling methods. We elaborate on two of the main categories of these methods, namely, uniform sampling and random walk.

Uniform sampling includes algorithms that choose the sample, which is a set of users (nodes), completely randomly without considering the friendship connections (edges) between users. Applying this sampling method requires global knowledge of the ID space of the users, so valid IDs can be randomly chosen [19, 28].

The random walk (RW) category contains a variety of algorithms where we walk from a user to one of their friends randomly [28, 30]. In classical RW, the next user u_{i+1} is selected from the friends of the current user u_i with the probability $p = 1/|friends(u_i)|$.

3 RELATED WORK

In this section, we first discuss related work from the field of PPARM and second, to cover the aspect of efficiency, ARM.

3.1 PPARM

PPARM can be classified into: **(1)** hiding sensitive rules and **(2)** cryptography-based approaches.

Hiding sensitive rules. Rules containing sensitive data about a person should be hidden. For that, datasets are commonly anonymized before mining by distortion/perturbation [10], blocking [13] up to deleting the sensitive attributes completely [11, 26]. Such modifications naturally lead to loss of information, consequently, decreasing the quality of the mined rules: the application of perturbation [10] on the breast-cancer database [21] led to the creation of 14% false rules and the loss of 28% non-sensitive rules [1].

An expectation-maximization algorithm to reconstruct the distribution of the original data can partially reduce this loss [2]. Other rule-hiding techniques [9, 36] constrain the support and confidence thresholds of the rules. Such constraints minimize the data incurred by the mining process in a way that omits the sensitive attributes from the output rules. However, constraining the support and confidence thresholds can leave an impact on all the mined rules, not just the sensitive attributes. Hence, the main challenge—again—is to not lose insensitive rules. Approaches balancing data utility and privacy are appearing recently: Kalyani et al. [16] use a particle swarm optimization algorithm to select and alter the transactions of the database to minimize the number of insensitive rules that are lost.

Yet, anonymization techniques can—inherently—only maintain partial properties of the complete dataset. In addition, more challenges remain in datasets obtain from OSNs, e.g., the relation between users rendering the anonymization even more complex.

Cryptography-based. ARM approaches based on cryptography usually assume ARM in distributed environments, i.e., distributed data. Here, every user controls their own data and does not wish to disclose it to other users.

A leading [32] approach [17] for horizontally partitioned data is based on a secure union and sum: first, local (per user) frequent itemsets (FIs), i.e., itemsets with high support, are collected and combined via a secure union. Next, the approach uses a secure sum to identify FIs also meeting the global support. The efficiency and scalability of this protocol are questionable since the protocol requires heavy cryptography operations (commutative encryption, oblivious transfers) and high costs in terms of messages. The secure union alone requires every user to encrypt the itemsets of every other user.

Some efficiency improvements, e.g., by using secure multi-party computation (SMC) for itemset intersections [32], the usage of Elliptic-curve cryptography to reduce the key size [6] as well as only partially distributed techniques [6], exist.

Overall, cryptography-based approaches cannot be applied to all application scenarios and at scale as, despite many improvements, the communication costs remain high and as the assumptions prefer a rather low distribution with only homogeneous itemsets.

3.2 Efficient ARM

We suggest that the efficiency of PPARM protocols can be improved by using efficient ARM techniques, therefore, we investigate these techniques. Kotsiantis et al. [18] classified the efficient ARM techniques into four categories:

- Reducing the number of passes over the database.
- Sampling the database.
- Adding extra constraints on the structure of rules.
- Parallelization.

Reducing the processed data in ARM is an intuitive practice to improve the privacy of the users. Only part of the data is processed, while the rest is excluded, thus, protected. Data reduction is enforced by the General Data Protection Regulation (GDPR) for this purpose. Sampling the data achieves data reduction. In addition, considering the distributed nature of our problem environment, sampling the data means reducing the number of involved users in PPARM, which in turn reduces the communication cost. For the previous reasons, we focus on the sampling techniques here.

Toivonen [33] proposed to sample the dataset randomly and find all the FIs in the sample. Since the sample was taken randomly, mostly not all the FIs in the database would be found in the sample. To obtain the missed FIs, the negative border of the found FIs is used to derive more FIs. Finally, all the FIs are verified by checking whether they are frequent with respect to the rest of the database.

As the validity of the sample is determined by two characteristics: the size of the sample and the quality of the sample. Parthasarathy [27] and Chuang et al. [8] suggested algorithms based on progressive sampling to determine the needed sample size to achieve association rules with the desired accuracy.

Zhang et al. [38] proposed reducing the database size in three steps. First, remove infrequent items (i.e., reduce the number of columns). Second, take a sample from the remaining database (i.e., reduce the transactions/rows). Third, cluster the transactions of the sample in granules.

Wang et al. [37] use a domain-specific ontology to mine association rules from a large volume of data. The ontology is used to reduce the number of items (i.e., columns) in the database. Mainly, the FIs are generated ignoring the itemsets that contain attributes that have no semantic relation according to the ontology. More sampling techniques can be found in an analysis study by Chakaravarthy et al. [7].

All the aforementioned approaches tackle the problem of efficient ARM for centralized databases, which does not apply to our setting of distributed data. In addition, the privacy aspect is not considered in most contributions.

4 OVERVIEW

In this work, our goal is improving the efficiency of the PPARM approaches to be applied in OSNs. We propose to sample the users to reduce the number of participant entities in the PPARM process. Thus, reducing the overall communication and computational costs. The data of the sampled users is collected and used to build the association rules in a privacy-preserving manner.

To obtain a sample of users, we adapt multiple sampling strategies to the domain of PPARM by using the social graph of users, identify the best sampling strategy, and elaborate on suitable optimizations and parameters settings to best suit efficiency or privacy respectively.

We use an adapted version of the random walk (RW), where the visited users decide locally and privately whether they become part of the sample; this walk is called anonymous random walk (ARW). Using the ARW eliminates the correspondence between individual users and their data, thus, neither the provider nor the users are able to know a particular user's itemsets. By that, we fulfill the privacy requirements i. and ii. presented in Section 2.1.

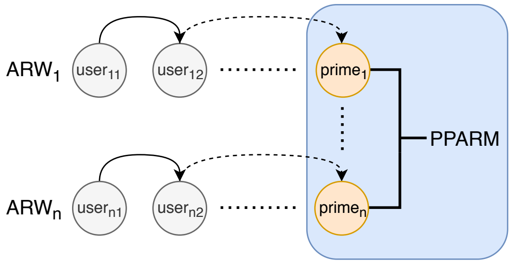


Figure 1: Overview of the sampling process.

We run multiple ARWs as shown in Figure 1. In each ARW, the data of a set of users is collected and stored by the last user in the walk, we refer to as a *prime user*. After applying multiple walks, the prime users can communicate and perform PPARM collaboratively to mine the FIs. Applying PPARM on a sample of the data can lead to some false FIs. Therefore, the FIs need to be verified later by all the users in case full-precision results are required.

In the following, Section 5 introduces the sampling algorithm ARW, then the process of PPARM and verification are presented in Section 6.

5 SOCIAL GRAPH-BASED SAMPLING

In this section, we discuss the basics of sampling for ARM in general, then using social graphs as a base for this sampling. Lastly, we provide more details about our approach.

Sampling for ARM. Applying PPARM for all users U considering all their data D is quite expensive in terms of communication and computation. Therefore, we sample the users and apply PPARM on the data sample S to find the frequent itemsets (FIs) that potentially represent all the user data. An optimal sample is a sample S that contains exactly the FIs that found in the complete dataset D; $FI_S = FI_D$. Obtaining such a sample is challenging as sampling by nature can lead to a loss in accuracy.

ARM builds a set of rules based on itemsets that occur frequently (FIs). The minimal support threshold to consider an itemset as FI is denoted by θ. As we are concerned about privacy and efficiency, we aim to obtain a good sample that includes frequent itemsets without disclosing information about these itemsets or performing highly complex calculations. Data sampling in our case is equivalent to sampling the users. Thus, we need an indicator that helps detect the users who have itemsets frequently occur in D, i.e., common between users. Including these users in the sample will emphasize the quality of the sample.

Social graphs. Users in OSNs are influenced by each other for several reasons [31]. The influence strength varies from a topic to another. For example, the work colleagues have an influence on each other's work-related interests. Friends are more likely to influence each other, thus, emphasizing the similarity between them. That conforms with the theory of homophily, which explains the tendency of individuals to associate themselves with others who are similar to them [23].

The ability of users to influence others varies from a user to another based on various factors. One of the main factors is the number of friends a user has [4]. Users with a high number of friends (connections) are able to reach a large audience. Thus, they are more likely to have a bigger influence in the OSN comparing with isolated ones; we refer to such users as *influencers*. An influencer can influence other users, for example, by bringing their attention to a particular item and making them interested in it. 49% of Twitter users rely on recommendations from influencers [35]. Consequently, users tend to imitate the itemsets of the influencers to some extent. As a result, we assume that the items of the influencers can be considered good representatives of a large number of influenced users. Therefore, considering the influencers in the sampling process can promote the quality of the sample. The influencers can be obtained from a social graph by looking for highly connected users.

Sampling methods. Next, we discuss the feasibility and efficiency of common sampling approaches, namely, uniform and random walk (RW). The uniform sampling chooses the sample randomly based on uniform distribution, i.e., all users have the same probability to be chosen. The RW sampling walks from a user to one of their friends randomly [28, 30]. Each visited user adds their data to the sample, then sends it to the next selected user.

Considering the efficiency of the sampling method is important as we deal with large-scale graphs such as OSNs. RW is more efficient than uniform sampling [29], which is costly due to the fact that the ID space is sparsely populated in many of the dominant OSNs. For example, the cost of uniform sampling in Flickr is $c = 77$, where c is the average number of IDs queried until one valid ID is obtained [29]. To avoid such a high cost, a global knowledge of the ID space is required, which is only available in the hands of the provider. Applying the sampling by the provider may threaten the privacy of the users who are part of the sample. Aside from the efficiency, RW can represent highly connected users (influencers) better than the uniform sampling, because RW is biased to highly connected nodes, while uniform sampling does not consider the connections between users. As discussed earlier, having influencers in the sample can improve the quality of the sample. However, in RW, all the users who are selected through the walk contribute with their data. Thus, each of these users knows that the previous and next users' data is part of the sample, which is a drawback from a privacy perspective.

To compensate for this risk, we propose using the anonymous random walk (ARW) [39]. In the ARW, each of the selected users decides locally whether to contribute with their data or not. Thus, not all the users who participate in the ARW are necessarily part of the sample. By that, the provider and other users (including prime users) cannot know which itemsets belong to which user, and whether a specific user has added their itemsets to the sample.

The PPARM approaches[17, 32] tackle the problem in distributed environments where there is a limited number of entities. The dataset is horizontally partitioned among the entities, i.e., each entity has a set of transactions (rows) of the dataset. In our problem setting, each user has their own data (row) only to preserve privacy. To apply the known PPARM approaches in OSNs, we need to collect the user data in multiple entities. Then, these entities can establish PPARM process collectively with each other. To form such entities, we perform multiple ARWs. The last user of each walk, called a *prime user*, has the data of a set of users. Thus, the prime users can apply PPARM to find the frequent itemsets.

The ARW in more detail is described in Algorithm 1, which we present briefly in the following. Each user starts an ARW with a predefined initializing probability p_{in}. Based on the sampling rate and the number of the walks, p_{in} is defined. By this practice, the provider cannot know which of the users started an ARW. Each visited user decides to add their data T with a contributing probability p_{co}, then sends the sample to the next user. The next user is chosen randomly from the visited user's friends. Multiple ARWs are performed simultaneously; each walk collects sub-sample. One user can add their data only to one sub-sample, so we avoid redundant data. The last user in each walk is announced as a prime user. The proposed algorithm uses two parameters that need to be specified: p_{in} and p_{co}.

Initializing probability p_{in}. Increasing p_{in} leads to more walks with a shorter length. That means, the PPARM is performed by more entities and, thus, increases the communication overhead. Small p_{in} implies a small number of prime users, which improves efficiency in PPARM while increases sampling communication overhead. However, long walks—more users take part in the walk—provides better privacy than short ones because the users who participate with their data will be hidden within bigger groups of users.

Contributing probability p_{co}. Having $p_{co} = 0.5$ leaves the adversary with only random guess abilities to find out if a specific user's data is part of the sample. Increasing $p_{co} \geq 0.5$ implies that the majority of the users visited by the walk add their data to the sample, which means less privacy. On the contrary, smaller p_{co} improves privacy but increase the length of the walk, thus, the communication overhead.

6 ARM & VERIFICATION

After collecting the data sample in the prime users, they can perform PPARM together. In this section, we discuss the implications of our sampling method on the accuracy and privacy aspects of one of the most accepted PPARM approaches [17] in the research community. We present two stages of the process: (1) finding the FIs and (2) verifying the FIs.

6.1 Finding FIs

A peer-to-peer approach for distributed PPARM was proposed in [17]; we refer to this approach as *UNIFI-KC* (Unifying lists of locally Frequent Itemsets— Kantarcioglu and Clifton). In this approach, a set of entities participate in mining the FIs collectively. The approach goes through two steps. First, each prime user mines the FIs locally, then all the local FIs are collected using secure union algorithms [17]. Second, the prime users calculate the support of the local FIs in respect to the whole dataset via a secure sum to find the global FIs.

From a privacy perspective, sampling the users naturally protect the privacy of the users who are not part of the sample. The data of the users in the sample is protected by using the ARW—as discussed in the previous section—and by using PPARM [17], which protects users from linkage attacks in the semi-honest model.

Algorithm 1 Anonymous Random Walk

```
Require: $U$ is the set of all users, $SR$ is the sampling rate, $S_k$ a
    sub-sample collected by the ARW no. $k$,
at each user $u_k$:
$p_{in}$ is predefined probability: $0 \leq p_{in} \leq SR$
$u_k$ randomly generates $0 \leq p_k \leq 1$
if $p_k < p_{in}$ then
    $u_k$ starts an ARW
    $u_c \longleftarrow u_k$
    let $S_k = \emptyset$
    $subSampleSize \longleftarrow SR/p_{in}$
    while $|S_k| \leq subSampleSize$ do
        if $u_c$ did not contribute to any sample before then
            $p_{co}$ is predefined probability: $0 \leq p_{co} \leq 1$
            $u_c$ randomly generates $0 \leq p \leq 1$
            if $p < p_{co}$ then
                append $T_c$ to $S_k$
            end if
        end if
        $u_c$ randomly selects $u_n \in friends(u_c)$
        $u_c$ sends $S_k$ to $u_n$
        $u_c \longleftarrow u_n$
    end while
    $u_c$ is announced as a prime user
end if
```

Regarding the accuracy of the results, we compare our approach with UNIFI-KC, where the data is not manipulated, thus, the detected FIs are the same as if a regular ARM has been applied. The accuracy of our approach depends on the parameter assignments of the sampling rate *SR*, initializing, and contributing probabilities p_{in}, p_{co}, respectively. Section 7 provides more insights into these parameters.

6.2 Verification

The application of ARM on sampled data S can lead to missing FIs, i.e., some FIs in D are found not frequent in S; $\exists X \in FI_D, X \notin FI_S$. To compensate this case, we lower the support threshold for the sample $\theta_S < \theta$ [33]. However, this may lead to generate false FIs, i.e., itemsets found to be frequent in S but not in D; $\exists X \notin FI_D, X \in FI_S$. To remove false FIs, we perform a verification.

In this phase, FIs FI_S is passed to all users, who then calculate the global support of FI_S. The itemsets that do not exceed the support threshold θ are excluded. The verification process guarantees that the precision rate of the itemsets found based on our sampling approach goes to 1.0.

It should be noted that FI_S does not threaten the users' privacy as it is an aggregate of all sampled users. The support counter can be processed in an efficient (as opposed to related PPARM approaches) and privacy-preserving manner. For that, we use a modified version of the secure sum protocol [17]. Algorithm 2 shows the steps of the protocol. User u_1 starts the protocol by generating random integers and assign them to occurrence counters of all the itemsets in FI_S. User u_1 increments the occurrence counter for itemset X if X in their data. Next, the FI_S and the occurrence counters are sent to the next user, which in turn checks their data and increments counters if needed. This ring ends up in the user u_1 who subtracts the random values, then calculates the actual global support for each itemset in FI_S.

Algorithm 2 FIs Verification

```
Require: $FI_S$ frequent itemsets in sample $S$.
at user $u_1$:
for all $X \in FI_S$ do
    generate random integer $r_X$
    $X.occurrence \leftarrow r_X$
    if $X \subseteq T_1$ then
        $X.occurrence++$
    end if
end for
send $FI_S$ and occurrence counters to next user $u_k$
at user $u_k$:
for all $X \in FI_S$ do
    if $X \subseteq T_k$ then
        $X.occurrence++$
    end if
end for
after passing all users, at user $u_1$:
for all $X \in FI_S$ do
    $s_D(X) = (X.occurrence - r_X)/|D|$
end for
```

7 EXPERIMENTS RESULTS

This section evaluates the correctness of the identified FIs, as well as the efficiency of the proposed sampling-based approach. We discuss the efficiency w.r.t. the communication cost of the sampling, PPARM, and verification. First, we explain the setup of our experiments based on real-world data, followed by a description of the dataset preprocessing and pre-analysis. Sections 7.4 and 7.5 then deal with the FIs correctness and efficiency.

7.1 Experimental Setting

Our sampling based on social graphs requires datasets that contain the social graphs of the contributing users in addition to the itemsets. However, such graphs are often omitted due to privacy and intellectual property concerns; experiments based on exclusive agreements with OSNs [20] are, therefore, not reproducible.

We use three open real-world graph datasets collected from Flickr (social photo sharing), Orkut (OSN), and Livejournal (OSN/diary) [24]. These networks allow users to create and join interest groups. The datasets contain the interest groups to which each user belongs and the social graphs (friendship connections between users). Table 1 summarizes the properties of these datasets.

We compare our approach ARW with uniform sampling as a base-line. The uniform sampling was used in several approaches to improve the efficiency of ARM [27, 33].

We use Python 2.7 to implement the sampling methods, and the *MLxtend* library to apply ARM. In detail, we apply the Apriori algorithm [3] on the dataset D and sample S. The FIs found in both D and S are referred to as FI_D and FI_S, respectively.

Datasets	Flickr	Orkut	Livejournal
# Users (10^6)	1.8	3.1	5.3
# Connections (10^6)	22	223	77
Average degree	24	144	29
# Interest groups (10^6)	0.103	8.7	7.4

Table 1: Overview of datasets

We consider four important parameters for the experiments:

ARM–Support threshold θ (complete dataset). Selecting θ is application-specific and adjusted interactively until interesting FIs are discovered. We empirically chose a variety of values that produce sufficient numbers of FIs (≥ 100).

ARM–Support threshold θ_S (sample). As ARW does not guarantee optimal samples, i.e., the samples do not necessarily contain all itemsets with the same occurrence rate as the complete dataset, we set $\theta_S \leq \theta$ as suggested in [33]. For the experiments, we use a range of values for θ_S and observe the correctness of FIs. We adjust the range for θ_S such that the whole spectrum of the recall and precision rates from 0 to 1 can be observed.

ARW–Initializing probability p_{in}. The value of p_{in} controls how many ARW should be performed to collect the sample. For the sake of simplicity, we discuss here only the number of ARWs k. Ribeiro et al. [29] proposed using multiple RW (uniform random jump) for sampling graphs as an improvement for the classic RW. To achieve good samples, Ribeiro et al. suggest that the number of walks k should be

$$0.5 * avg(\text{outdegree}) < k < 2 * avg(\text{outdegree}) \qquad (2)$$

Where outdegree is the number of direct connections from a user to others. In our case, we consider indirect connections, thus, we deal only with the *degree* parameter, which is the number of connections of a user. In our experiments, we fix $k = avg(\text{degree})$.

ARW–Contributing probability p_{co}. We fix $p_{co} = 0.5$ in all the experiments unless we state otherwise. With this probability, an adversary can only randomly guess whether a visited user is part of the sample.

7.2 Dataset Preprocessing

We only consider connected graphs, i.e., users with at least one friend, for the experiments. Users with no interests are discarded as they have no data to contribute with.

As sampling is sensitive to the population size, we fetch slices of varying size from the datasets. Well-known sampling techniques, such as random node- and edge sampling, could be used here [19]; however, these techniques can lead to very sparse graphs, which do not represent the communities in the original graph.Therefore, we propose our own slicing algorithm as follows: let z be the slice size we desire. We randomly select a *userID* as a starting point for the slice. We take the users whose IDs belong to $[userID, userID + z]$. We build the social graph of these users, i.e., the connections between these users. We select the largest connected component g in this graph, as we focus on connected graphs. In case g contains fewer users than z, we add more users as follows. We select randomly a user $u \in g$, then we add randomly one of his friends $u_i \in friends(u), u_i \notin g$. We add all the connections that u_i has with other users in g. We repeat this process until g reaches the desired size. With this slicing technique, we achieve a connected graph that reflects the topology of part of the original graph.

7.3 Dataset Pre-analysis

The idea of using social graphs for sampling is based on the assumption that similar users (i.e., share similar itemsets) are more likely to be connected. Prior to our experiments, we conducted a study about the correlation between having same itemsets and the density of users in the Flickr dataset. The density of a set of n users is $density = |\text{connections}|/(n(n-1)/2)$, where $n(n-1)/2$ is the number of edges in a complete graph. Figure 2 shows that the density of the users who have same frequent itemsets is higher than the average destiny of same number of users chosen randomly from the graph, which supports our aforementioned assumption.

7.4 FIs Correctness

To evaluate the correctness of FIs found in the samples, we consider FI_D as a reference and calculate the recall and precision rates of FI_S for each sampling method: uniform and ARW. In addition, we compute the area under the PR curve (AUCPR) using the average precision score [5]. In each experiment, a single value for θ is used to mine FI_D, while a range of values for θ_S is examined to create the precision-recall curves for FI_S. Each experiment is repeated for 25 random slices of the same size.

For each slice, we perform the sampling, ARM, and the evaluation 25 times and calculate the average recall and precision rates of all the repetitions. In this section, we investigate the correctness of FIs considering different: (1) sampling rates, (2) population sizes, (3) graph topologies, and (4) contributing probabilities.

Sampling rates. We conduct experiments with different sampling rates $SR \in \{30\%, 50\%\}$ on slices of 10 000 users from the Flickr dataset. The average degree of users varies from 116 to 237, which indicates the number of ARWs for each slice. Figure 3a shows that decreasing the sampling rate leads to lower recall and precision for both ARW and uniform sampling; that is expected as a smaller sample is more likely to provide a less accurate representation of the original dataset. The difference of the average precision score between uniform AP_{uni} and ARW AP_{arw} decreases with a higher sampling rate. For $SR = 30\%$, $AP_{uni} - AP_{arw} = 0.07$, while for $SR = 50\%$, $AP_{uni} - AP_{arw} = 0.03$. However, ARW drops tangibly as we can see for $SR = 30\%$. The small SR leads to shorter walks, which may limit the walks to a few number of communities. As a result, the sample does not reflect a wide spectrum of users.

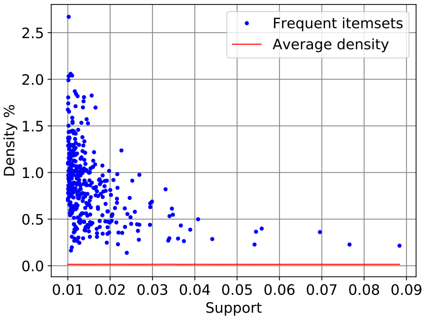


Figure 2: Efficiency related metrics.

Population sizes. We conduct further experiments to investigate the FIs correctness with different dataset sizes from the Flickr dataset. We choose two slices of data that contain 10 000 and 100 000 users and compare it with Flickr as a whole, which contains ~ 360 000 users after preprocessing. We used $SR = 50\%$. Interestingly, we observe in Figure 3b that our approach performs better as the population grows, where the number of users is closer to realistic scenarios. This supports the applicability of our approach in real-world OSNs.

Graph topology. Next, we take a look at the impact of the topologies. Therefore, we conduct experiments on slices of 100 000 users from the three datasets Flickr, Orkut, and Livejournal, with $SR = 50\%$ as before. The topologies' differences between the datasets can be summarized as follows: Flickr has a big central community and a limited number of sub-communities. While both of Livejounral and Orkut contain clearer communities. However, in Orkut, the communities are much more connected and users in general have more connections, this is evidenced by the high average degree 144 (See Table 1).

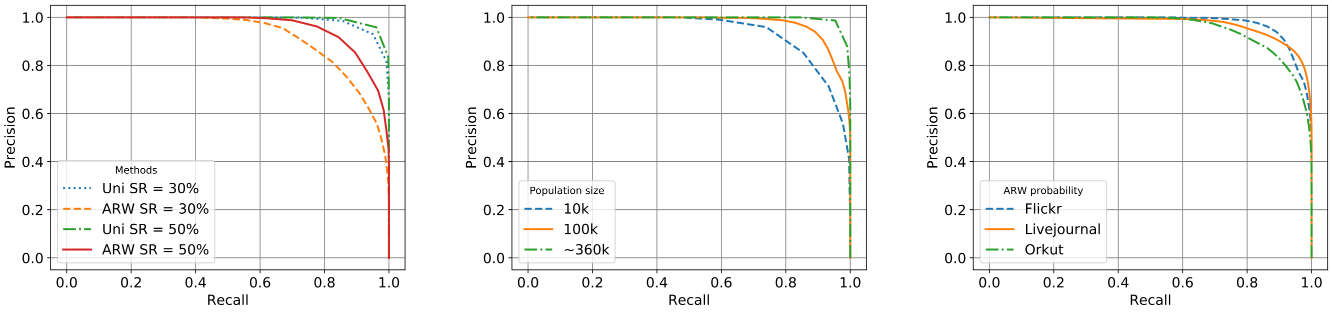
These differences may relate to the nature of the social network's purposes. Flickr and Livejournal are more specialized social networks, while Orkut is a general purpose one. In Figure 3c, we can observe that Flickr and Livejournal both perform better than Orkut. The reason can be that the types of relations in Orkut are more diverse than Flickr and Livejournal. Meaning, the friendships' connections are influenced by many factors as the OSN is general and contain groups from various fields and interests, while Flickr and Livejournal are about images and blogs only. We conclude that our sampling approach can be more effective in specialized OSNs.

Contributing probability. Performing ARW requires the specification of a data contribution probability for the users. We investigate the impact of a range of probabilities on the correctness of FIs. We perform ARW for three values of probability $p_{co} \in \{0.1, 0.5, 1.0\}$ on 10 000 users of Flickr. In the case of $p_{co} = 1.0$, the ARW renders a normal RW. The results show that the precision and recall improve very slightly with growing p_{co}.The average precision score grows by only 0.01 between $p_{co} = 0.1$ and $p_{co} = 1.0$. From the privacy perspective, reducing p_{co} improves the privacy as more users are visited, thus, the contributing users (and their data) are hidden within a bigger group of users. We conclude that it is possible to increase the privacy level in ARW by adjusting the p_{co} value without a tangible loss in the correctness of FIs.

7.5 Efficiency

In this section, we look at the efficiency of three processes sampling, PPARM, and verification.

Sampling. With the focus on the communication cost, we consider the number of messages as the main metric for the evaluation. In ARW, the number of messages needed to collect the sample is linear to the contributing probability p_{co}. A small p_{co}, e.g., $p_{co} = 0.1$, indicates more visited users and, thus, better privacy; However,



(a) ARW & uniform–Sampling rates

(b) ARW–Population sizes

(c) ARW–Topologies

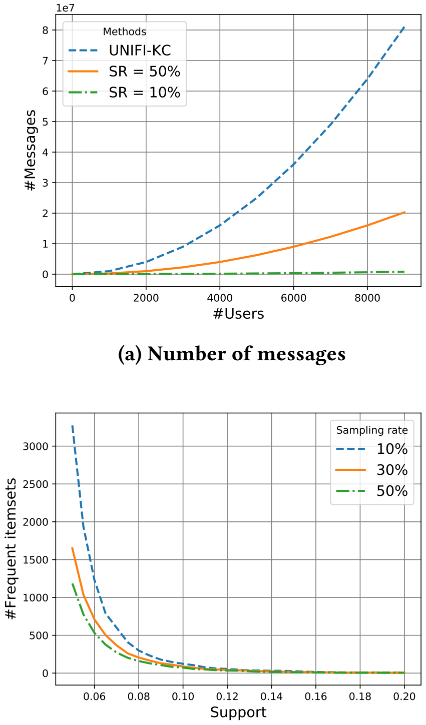
Figure 3: Precision-recall curves.

this increases the number of messages, for example, by five folds comparing with $p_{co} = 0.5$.

PPARM. In our problem setting, we assume a large-scale application with potentially millions of users, where each user has a small amount of data. The communication cost in terms of message tightly relates to the number of participating users n.*UNIFI-KC* requires $(n^2 + 2n - 3)(K + 1)$ messages in total [17]; where $K + 1$ is the number of iterations, which correspond to the size of the FIs. For the sake of simplicity, we consider one iteration $K = 0$. Figure 4a illustrates the message number reduction when sampling is used before applying *UNIFI-KC*. We can see that sampling rate $SR = 50\%$, which achieves high-quality FIs, can reduce the messages overhead by four folds.

Verification. We discuss here the number of FIs as a metric for computational cost. In Figure 4b, we show the number of generated FIs over varying sampling rates SR and supports θ_S.It is obvious that maintaining the same support threshold for a smaller number of users, the detection of FIs increases. Thus, more computation cost for the users and bigger messages to be passed between them for the verification. The number of FIs increases by 30% from $SR = 50\%$ to $SR = 30\%$. With lower SR, such as $SR = 10\%$, FIs increase by 125% comparing with $SR = 50\%$. However, we can see that the higher the support threshold, the less sensitive the FIs are to the sampling rate.

A lower sampling rate SR leads to efficiency improvement in PPARM but increases the cost of the verification phase as we detect more FIs, especially if we use a low θ_S. A calibration for these two parameters should be conducted for every application scenario to meet the scenario's requirements.



(a) Number of messages

(b) Number of frequent itemsets

Figure 4: Efficiency related metrics.

8 CONCLUSION

In this paper, we proposed a sampling-based approach to improve the efficiency and privacy of PPARM. Our approach uses the social graph in OSNs as a base for the sampling. This sampling approach improves the privacy over centralized ARM as the users remain in control of their data and as the data required to generate FIs is minimized. The sampling strategy is, therefore, also comparably efficient to common PPARM approaches while improving the correctness of mined FIs due to the better sample selection. We analyzed the impact of different sampling rates, population sizes, and topologies, on the quality of the sample. We tested our approach on three real-world OSNs's datasets. The results showed the feasibility of our approach in large-scale applications such as OSNs. Future work will focus on extending the base of the sampling to consider—in addition to social graphs—similarity graphs, which potentially improve the quality of the sampling process.

REFERENCES

[1] Doryaneh Hossien Afshari and Farsad Zamani Boroujeni. 2017. Using blocking approach to preserve privacy in classification rules by inserting dummy Transaction. *Journal of Soft Computing and Applications* 2017, 1 (2017), 44–52. https://doi.org/10.5899/2017/jsca-00073

[2] Dakshi Agrawal and Charu C Aggarwal. 2001. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 247–255.

[3] Rakesh Agrawal, Ramakrishnan Srikant, et al. 1994. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, Vol. 1215. 487–499.

[4] Eytan Bakshy, Jake M Hofman, Winter A Mason, and Duncan J Watts. 2011. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 65–74.

[5] Kendrick Boyd, Kevin H Eng, and C David Page. 2013. Area under the precision-recall curve: point estimates and confidence intervals. In *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 451–466.

[6] Harendra Chahar, B. N. Keshavamurthy, and Chirag Modi. 2017. Privacy-preserving distributed mining of association rules using Elliptic-curve cryptosystem and Shamir's secret sharing scheme. *Sadhana - Academy Proceedings in Engineering Sciences* 42, 12 (2017), 1997–2007. https://doi.org/10.1007/s12046-017-0743-4

[7] Venkatesan T Chakaravarthy, Vinayaka Pandit, and Yogish Sabharwal. 2009. Analysis of sampling techniques for association rule mining. In *Proceedings of the 12th international conference on database theory*. ACM, 276–283.

[8] Kun-Ta Chuang, Ming-Syan Chen, and Wen-Chieh Yang. 2005. Progressive sampling for association rules based on sampling error estimation. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 505–515.

[9] Elena Dasseni, Vassilios S Verykios, Ahmed K Elmagarmid, and Elisa Bertino. 2001. Hiding association rules by using confidence and support. In *International Workshop on Information Hiding*. Springer, 369–383.

[10] Aggelos Delis, Vassilios S Verykios, and Achilleas A Tsitsonis. 2010. A data perturbation approach to sensitive classification rule hiding. In *Proceedings of the 2010 ACM Symposium on Applied Computing*. ACM, 605–609.

[11] Alexandre Evfimievski, Ramakrishnan Srikant, Rakesh Agrawal, and Johannes Gehrke. 2004. Privacy preserving mining of association rules. *Information Systems* 29, 4 (2004), 343–364.

[12] Niall Ferguson. 2017. The False Prophecy of Hyperconnection: How to Survive the Networked Age. *Foreign Aff.* 96 (2017), 68.

[13] Vikram Garg, Anju Singh, and Divakar Singh. 2014. A survey of association rule hiding algorithms. *Proceedings - 2014 4th International Conference on Communication Systems and Network Technologies, CSNT 2014* (2014), 404–407. https://doi.org/10.1109/CSNT.2014.86

[14] The Guardian. 2018. Facebook to contact 87 million users affected by data breach. https://www.theguardian.com/technology/2018/apr/08/facebook-to-contact-the-87-million-users-affected-by-data-breach. [Online; accessed 11-Dec-2018].

[15] The Guardian. 2018. Huge Facebook breach leaves thousands of other apps vulnerable. https://www.theguardian.com/technology/2018/oct/02/facebook-hack-compromised-accounts-tokens. [Online; accessed 11-Dec-2018].

[16] G Kalyani, M V P Chandra Sekhara Rao, and B Janakiramaiah. 2017. Particle Swarm Intelligence and Impact Factor-Based Privacy Preserving Association Rule Mining for Balancing Data Utility and Knowledge Privacy. *Arabian Journal for Science and Engineering* (2017). https://doi.org/10.1007/s13369-017-2834-2

[17] Murat Kantarcioglu and Chris Clifton. 2004. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Transactions on Knowledge and Data Engineering* 16, 9 (2004), 1026–1037. https://doi.org/10.1109/TKDE.2004.45

[18] Sotiris Kotsiantis and Dimitris Kanellopoulos. 2006. Association Rules Mining: A Recent Overview. *Greece - Science* 32, 1 (2006), 71–82. https://doi.org/10.4103/0377-4929.94858

[19] Jure Leskovec and Christos Faloutsos. 2006. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 631–636.

[20] Dongsheng Li, Qin Lv, Li Shang, and Ning Gu. 2017. Efficient privacy-preserving content recommendation for online social communities. *Neurocomputing* 219, September 2016 (2017), 440–454. https://doi.org/10.1016/j.neucom.2016.09.059

[21] MultiMedia LLC. 1995. Breast Cancer Wisconsin (Diagnostic) Data Set. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29. [Online; accessed 11-Dec-2018].

[22] David McCandless. 2019. World's Biggest Data Breaches & Hacks. https://informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/. [Online; accessed 02-Jan-2019].

[23] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.

[24] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and Analysis of Online Social Networks. In *Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC'07)*. San Diego, CA.

[25] Miquel Montaner, Beatriz López, and Josep Lluís De La Rosa. 2003. A taxonomy of recommender agents on the internet. *Artificial intelligence review* 19, 4 (2003), 285–330.

[26] Stanley RM Oliveira and Osmar R Zaiane. 2002. Privacy preserving frequent itemset mining. In *Proceedings of the IEEE international conference on Privacy, security and data mining-Volume 14*. Australian Computer Society, Inc., 43–54.

[27] Srinivasan Parthasarathy. 2002. Efficient progressive sampling for association rules. In *null*. IEEE, 354.

[28] Bruno Ribeiro and Don Towsley. 2010. Estimating and sampling graphs with multidimensional random walks. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 390–403.

[29] Bruno Ribeiro, Pinghui Wang, Fabricio Murai, and Don Towsley. 2012. Sampling directed graphs with random walks. In *INFOCOM, 2012 Proceedings IEEE*. IEEE, 1692–1700.

[30] Daniel Stutzbach, Reza Rejaie, Nick Duffield, Subhabrata Sen, and Walter Willinger. 2006. Sampling techniques for large, dynamic graphs. In *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*. IEEE, 1–6.

[31] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. 2009. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 807–816.

[32] Tamir Tassa. 2014. Secure mining of association rules in horizontally distributed databases. *IEEE Transactions on Knowledge and Data Engineering* 26, 4 (2014), 970–983. https://doi.org/10.1109/TKDE.2013.41 arXiv:1106.5113

[33] Hannu Toivonen. 1996. Sampling large databases for association rules. In *VLDB*, Vol. 96. 134–145.

[34] Theja Tulabandhula, Shailesh Vaya, and Aritra Dhar. 2017. Privacy-preserving Targeted Advertising. (2017), 1–16. arXiv:1710.03275 http://arxiv.org/abs/1710.03275

[35] Twitter and Annalect. 2016. The value of influencers on Twitter. https://blog.twitter.com/marketing/en_us/a/2016/new-research-the-value-of-influencers-on-twitter.html. [Online; accessed 18-Apr-2019].

[36] Vassilios S Verykios, Ahmed K Elmagarmid, Elisa Bertino, Yücel Saygin, and Elena Dasseni. 2004. Association rule hiding. *IEEE Transactions on knowledge and data engineering* 16, 4 (2004), 434–447.

[37] Yongqing Wang and Yan Chen. 2012. ANew Association Rules Mining Method based on Ontology Theory. (2012).

[38] Zhongjie Zhang, Witold Pedrycz, and Jian Huang. 2017. Efficient frequent itemsets mining through sampling and information granulation. *Engineering Applications of Artificial Intelligence* 65 (2017), 119–136.

[39] Yingying Zhao, Dongsheng Li, Qin Lv, and Li Shang. 2018. A Scalable Algorithm for Privacy-Preserving Item-based Top-N Recommendation. *arXiv preprint arXiv:1811.02217* (2018).