This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2801798, IEEE Transactions on Network Science and Engineering

A Fair Mechanism for Private Data Publication in Online Social Networks

Xu Zheng[1,2], *Student Member, IEEE,* Guangchun Luo[1,*], Zhipeng Cai[2], *Senior Member, IEEE*

Abstract—Due to the tremendous growth of online social networks in both participants and collected contents, social data publication has provided an opportunity for numerous services. However, neglectfully publishing all the contents leads to severe disclosure of sensitive information due to diverse user behaviors. Therefore, there should be a thoroughly designed framework for data publication in online social networks that considers users heterogeneous privacy preferences and the correlations among participants. This work proposes a novel mechanism for data publication that achieves high performance while preserving privacy and guaranteeing fairness among users. To derive the optimal scheme for data publication is NP-complete. Thus we propose a heuristic algorithm to determine the contents to be published which takes advantage of the sets of sensitive contents for each user and the correlation among them. The theoretical analysis proves the effectiveness and feasibility of the mechanism. The evaluations towards a real-world dataset reveal that the proposed algorithm outperforms the existing results.

Index Terms—Online Social Networks; Data Publication; Privacy Preservation; Fairness

1 Motivation

The pervasive adoption of Online Social Networks (OSNs) has made this category of systems a pivotal component for knowledge discovery [1]. Large scale of contents are generated in OSNs, owing to both the huge number of participants and various dimensions of the available information [2], [3]. Meanwhile, users in OSNs are willing to publish their contents like ratings for restaurants or movies, which could be utilized by service providers to facilitate various kinds of services including content recommendation, friend recommendation, *etc* [4], [5]. However, the careless publication of such a huge size of contents may lead to severe disclosure of users' sensitive information. For example, the sexual orientation may be inferred from the movies watched by a user [6], and the residential area is usually close to the frequently visited restaurants [7]. Therefore, a novel framework for data publication in OSNs is designed in this work, which considers both privacy preservation and quality of service for heterogeneous users.

In fact, the large scale of social data makes it possible for users to publish partial contents [8], [9], while service providers can still guarantee qualified services. Service providers can utilize the contents gathered from multiple users to facilitate services. However, the following new challenges emerge when a partial publication strategy is utilized.

Firstly, considering the pervasive existence of both direct and latent sensitive information, users may have heterogeneous preferences on sensitive information. Their attitudes could range from concealing all the contents to totally careless about private information or even unconscious. As a result, each user should have a customized publication scheme which can properly preserve privacy as expected.

Secondly, users are usually correlated with each other in a service. For instance, to recommend restaurants to a target user, a service provider selects the visited restaurants from the users who share similar experiences with the target user. Therefore, a partial content publication scheme should also properly choose the contents based on the correlation among users.

More specifically, partially published contents should guarantee high quality of service, which can be measured form two aspects. On the one hand, a majority of users should receive qualified services based on the published contents. This requirement is essential since users are correlated with each other, and some sensitive contents must be concealed, which means it is usually infeasible to serve all users simultaneously. Take content recommendation as an example. The published contents must help with providing a sufficient number of recommended contents for as many users as possible. On the other hand, the published contents should provide a fair service among users. As some users prefer to benefit more and contribute more to a service, it is necessary to provide better services to the more active users. Again, take content recommendation as an instance. The published contents should priorly serve users sharing their own contents without limitation. In summary, a content publication scheme should take into account privacy, utility, and fairness simultaneously.

Unfortunately, these challenges have not been well addressed by the previous works. The existing solutions mainly consider the scenario with identical preference on privacy preservation, or the most stringent requirement among all users. These solutions, such as differential privacy [10] and k-anonymity [11], obfuscate contents or statistical results before publication. Some works allow personalized prefer-

1 X. Zheng and G. Luo are with School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, China, 611731. Email: xzheng7@student.gsu.edu, gcluo@uestc.edu.cn.
2 Z. Cai and X. Zheng are with Department of Computer Science, Georgia State University, 25 Park Place, Atlanta, GA, 30302, USA. Email: zcai@gsu.edu
** G. Luo is the corresponding author.*
Manuscript received May, 25, 2017; revised June, 14, 2017.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2801798, IEEE Transactions on Network Science and Engineering

ence on privacy. However, such works do not thoroughly consider the correlation among users. Most of them simply provide service independently for each user, like perturbing location information according to the user-specified noise level [12]. Applications like crowd-sensing [13] partially consider the correlation among users. However, they mainly treat each user as an independent individual to work on some common tasks, and are designed to achieve an equilibrium state.

This paper mitigates the gap by designing a novel mechanism for content/data publication in OSNs. The mechanism properly preserves the private information for each user. Meanwhile, the mechanism considers quality of service in terms of privacy, global utility and fairness among users. The published contents can be utilized by third-party servers for various subsequent services. As far as we know, this is the first study on content publication in OSNs which integrates correlation among users, users' heterogeneous privacy preferences, and fairness among users.

For privacy preservation, the proposed framework assumes adversaries can fully access the published contents and attack by running some inference functions on the data. The inference functions range from the ones searching for directly sensitive contents to the ones extracting latent private information. For example, an adversary may look for the restaurants near a hospital, or infers a user's residential area by investigating the frequently visited regions. Our framework thwarts all the attacks by deriving for each user all the sets of sensitive contents leading to the disclosure of private information and avoiding publishing them. Furthermore, users may select heterogeneous privacy preferences by setting different thresholds on inference functions, which leads to different number and composition of sensitive content sets.

For quality of service, our framework is mainly designed for one type of applications, *i.e.*, content-based recommendation applications. This category of applications include recommendation of interested contents, friend recommendation, *etc*. They firstly derive the pairs of users who share similar contents, and then recommend their unique contents to each other. Our framework guarantees that a maximal number of users can receive satisfied recommendation results, which means the recommendation results are comparable to the scenario where all the contents are published. Furthermore, our framework maintains fairness among users. The activeness and priority are measured according to each user's privacy preference. A more flexible preference setting indicates more active user participation. Our framework priorly serves the users with more flexible privacy settings among the users with similar scale of published contents. To derive the content publishing scheme which serves the maximum number of users is proved to be NP-complete. Therefore, this work proposes an effective heuristic algorithm and validates it towards a real-world dataset. Our contributions are summarized as follows:

1) A novel content publication model which captures correlation among users, heterogeneous privacy settings, and service fairness is proposed.
2) We introduce a new definition for fairness which relates user priority to privacy setting.
3) An algorithm is proposed to strictly preserve sensitive information, balance performance among users, and achieve a high global utility.
4) The corresponding analysis validates the performance of the proposed algorithm and emphasizes the feasibility of our framework.
5) Extensive experiment results towards a real-world dataset demonstrate the effectiveness of the algorithm.

The remaining of the paper is organized as follows. Section II reviews the related works. Section III presents the problem formulation. Section IV introduces the overview of the framework and the algorithm for content publication. Section V discusses the derivation of sensitive content sets and some corresponding applications. The experiment results shown in Section VI validate the performance of the algorithm. Section VII concludes the paper.

2 Related Works

Data publication for OSNs has been extensively studied during the past decades. Not surprisingly, many proposed solutions take privacy issues into consideration. For example, some works [14] consider the exposure of information like locations. They mainly differ in the definitions of sensitive information and the knowledge owned by adversaries. For instance, the study in [8] assumes adversaries own the optimal attacking strategy, and proposes some corresponding countermeasures for content publication.

There are some works considering user correlation. For example, the work in [15] proposes a mechanism where participants buffer some local information for each other, and only upload the unique information. This mechanism can decrease the scale of the published contents for each user to conceal sensitive information. Zheng *et al.* proposed a framework [16] where users publish reviews together for local business. It takes the advantage that users may comment on the same local business. The proposed scheme in [17] utilizes the uploaded information to generate a private meeting location for a whole community. Furthermore, applications like crowdsensing consider both user correlation and privacy [18]. They are mainly diverse in the factors involved in the platform. For example, citation [19] investigates how to utilize the feedbacks to update the incentive, and citation [20] studies the aggregation tasks together with privacy preservation. However, they mainly regard users as independent individuals and assume users are selfish. In the above works, users concern more on individual profits, while our work considers the scenario where users help each other to promote satisfiable services.

For privacy preservation, differential privacy [21] is currently one of the most popular techniques. This technique allows an adversary to know all the users' information except the target user, which is the same as our assumption. Differential privacy also considers user correlation during data publication, as it provides statistical results combining data from all users. For example, mining of frequent graph pattern under differential privacy is studied in [22]. Other works [23] [24] generalize, add noise and release data for general purposes. Publication of serial data under differential privacy is studied in [25] and [26]. However, they can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2801798, IEEE Transactions on Network Science and Engineering

only preserve privacy for identical or the most strict user preference.

Finally, fairness among users is considered from several aspects. Han *et al.* studied the influence maximization problem in OSNs by considering both the total size of coverage and the coverage for different types of users [27] [28] [29], including the consideration on the time delay, the dynamic OSNs, and the unique seed users. However, their works focus on the links among users, while our work considers content publication. Fairness is also considered in device-to-device communications [30], and some crowdsourcing or spatial crowdsourcing systems [31] [32] [33]. These works consider the fairness from different aspects, including their correlation with safety and incentive [31], the fairness on the distance of movement [32], and the correlation between the fairness and the loyalty [33]. All these works assume users are identical and compete or cooperate for benefits, while our work introduces heterogeneous privacy preferences and defines the priority levels accordingly.

3 Problem Formulation

We consider data publication for typical OSNs. In this circumstance, the platform will first get authorization and preferences on privacy from users and then publish their contents to some third-party servers. The published contents can be utilized by third-party servers for knowledge discovery to provide further services. Assume there are totally N users, denoted as $U = \{u_1, u_2, \cdots, u_N\}$. Each user u_i has a set of contents $C_i = \{c_{i1}, c_{i2}, \cdots, c_{iK_i}\}$. The contents could be a list of restaurants visited by a user, or movies once watched. All the candidate contents form a content pool $C_0 = \{c_{01}, c_{02}, \cdots, c_{0K_0}\}$.

System Utility: As our framework is mainly designed for content-based recommendation applications, it is critical to retain a sufficient number of similar users with each target user in the published data. In this circumstance, similar users refer to the ones who own some identical contents with the target user as well as some different contents. For simplicity, we note the common contents between two users as ***similar contents***, and the different ones as ***diverse contents***. Furthermore, the combination of the published different contents from all the similar users determine the quality of service for the target user.

Formally, assume I_{ij} is the indicator for the publication of c_{ij}, where $I_{ij} = 1$ means c_{ij} is published, and $I_{ij} = 0$ otherwise. Then two users u_i and $u_{i'}$ are similar towards the published data when

$$D_{ii'} = \sum_{j \le K_i, j' \le K_{i'}, c_{ij} = c_{i'j'}} I_{ij} I_{i'j'} \ge \delta, \tag{1}$$

where δ is the threshold to determine the similarity of two users. Equation (1) means two users are similar when they share no less than δ common contents in the published data. Meanwhile, the service quality for a target user is dominated by the exclusive contents published by all her similar users, which is evaluated as follows:

$$Q_{i'} = \frac{|\bigcup_{i=1}^{N} \{c_{ij} | I_{ij} = 1, D_{ii'} \ge \delta, 1 \le j \le K_i, c_{ij} \notin C_{i'}\}|}{|\bigcup_{i=1}^{N} \{c_{ij} | D^0_{ii'} \ge \delta, 1 \le j \le K_i, c_{ij} \notin C_{i'}\}|}, \tag{2}$$

where $u_{i'}$ is the target user, $|\cdot|$ refers to the number of contents in the set, and $D^0_{ii'}$ is the number of common contents when u_i and u'_i publish all their contents. Based on this measurement, user u_i is covered or successfully served when

$$Q_i \ge \gamma, \tag{3}$$

which means the size of diverse contents in the published data covers more than γ of all diverse contents.

Privacy: In this work, adversaries could be any third-party server which can access the published data. Adversaries are assumed to be honest but curious, *i.e.*, they extensively extract sensitive information from the published data. More specifically, an adversary holds an inferring function $f(\cdot)$, which takes the published contents as the input, and is used to infer the sensitive information of a user. For example, $f(\cdot)$ may check the existence of a police office, the number of restaurants near a hospital, or the likelihood of getting a disease. Based on the function, each user u_i selects a preference factor θ_i on the sensitive information. Denote the published data for user u_i as C'_i, then u_i's privacy is preserved when

$$\sum_{k=1}^{m} |Pr(f(C_i) = k) - Pr(f(C'_i) = k)| \ge \theta_i. \tag{4}$$

Therefore, a larger θ_i means u_i concerns more on her sensitive information, and shows less activeness in joining data publication.

Fairness: Besides utility and privacy, user fairness is another concern in content publishing. There are various definitions of fairness. In this work, we utilize privacy preference as a reference to determine users' priority levels accordingly. This measurement is intuitive since users with more flexible preference settings are more willing to participate in and contribute to content publication, which means they deserve a better service. According to this evaluation method for priority levels, fairness is defined as the permission of selfishness among users, which means users with higher priority levels are allowed to be selfish towards users with lower priority levels. We formally define fairness as follows.

Definition 1 (Fairness). Assume two users u_i and u_j have different preference factors, saying $\theta_i \ge \theta_j$. Then the published contents are fair for u_i and u_j when at least one of the following three conditions holds:

1) u_j could still be covered without u_i's published contents;

2) u_i's published contents are used to cover u_k and u_k has a higher priority than u_j;

3) u_i and u_j are not similar users.

Otherwise, u_i must contribute to the coverage of u_j. The term "coverage" between two users means 1) u_i is similar to u_j and, 2) u_i owns some different contents for u_j in the published data.

Finally, the objective is to design a content publication scheme which maximizes the number of covered users, while preserving sensitive information under the preference factors and guaranteeing user fairness. We formulate the problem as follows:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2801798, IEEE Transactions on Network Science and Engineering

$$\max \quad \sum_{i=1}^{N} G_i \tag{5}$$

$$\text{s.t.} \quad I_{ij} \in \{0,1\} \qquad \forall i, 1 \le j \le K_i \tag{6}$$

$$G_i = 1 \qquad \forall i, Q_i \ge \gamma \tag{7}$$

$$G_i = 0 \qquad \forall i, Q_i < \gamma \tag{8}$$

$$F(C_i') \ge \theta_i \qquad \forall i \in \{1, 2, \cdots, N\} \tag{9}$$

$$C_j' \bigcup Div(C_i) \ne \phi \quad \forall u_j \text{ contributes to } u_i, \tag{10}$$

where G_i indicates whether u_i is covered, $F(C_i') = \sum_{k=1}^{m} |Pr(f(C_i) = k) - Pr(f(C_i') = k)|$, and $Div(C_i) = \bigcup_{i'=1}^{N} \{c_{i'j} | D_{i'i}^0 \ge \delta, 1 \le j \le K_i', c_{i'j} \notin C_i\}$. Condition (6) means each content is assigned a decision on whether to be published. Conditions (7) and (8) refer to the coverage of a user. Conditions (9) and (1) denote user privacy and user fairness. The list of notations is shown in Table 1.

TABLE 1: List of Notation

Notation	Explanation
$U = \{u_1, u_2, \cdots, u_N\}$	Set of users
$C_i = \{c_{i1}, c_{i2}, \cdots, c_{iK_i}\}$	Contents from user u_i
$C_0 = \{c_{01}, c_{02}, \cdots, c_{0K_0}\}$	Candidate set of contents
I_{ij}	Indicator c_{ij} is published by u_i
$D_{ii'}$	Common contents by u_i and $u_{i'}$
δ	Threshold on similar users
$Q_{i'}$	Ratio of recommended contents
γ	Threshold for successful service
$f(\cdot)$	Inference function
θ_i	Privacy Preference of u_i
SS_i	Sensitive set for u_i
$V_i = \{v_{i1}, v_{i2}, \cdots, v_{iK_0}\}$	Diverse content set for u_i
$CV_i = \{cv_{i1}, cv_{i2}, \cdots, cv_{iK_0}\}$	Covered diverse content set for u_i

4 Solution

In this section, we first analyze the complexity of the proposed problem, then introduce the algorithm for determining the publication scheme in OSNs. The algorithm is called ***Priority-Based Content Publishing Algorithm*** (PBCP for short).

4.1 Complexity

The proposed problem is an NP-complete problem. We prove this by reducing the maximum independent set problem to ours. Assume there are two users in an OSN, and they are similar to each other. According to the inferring function, each user has a set of tuples each composed of contents, and the publication of both contents belonging to any tuple leads to privacy leakage. In this case, the tuples are considered as the edges between contents. Now given ratio γ on the coverage of diverse contents, our problem is the same as the decision problem of the independent set problem, which is NP-complete.

4.2 Data Structures

PBCP needs to maintain three data structures: sensitive sets, diverse vectors, and a dependent graph. The sensitive sets refer to all the combination of the contents for a user leading to the disclosure of sensitive information. A diverse vector records the set of contents and covered contents belonging to the diverse set of a user. The dependent graph indicates the similarity between users.

Sensitive Sets. Based on the function, each user has a set of tuples of contents that may lead to the disclosure of sensitive information, *i.e.*, based on these contents, adversaries can successfully infer some private information via the inferring function. A successful inference means the requested privacy preservation in Equation 4 is violated. Denote the sensitive content sets for u_i as $\{SS_i = ss_{i1} = \{c_{ij_1}, \cdots\}, \cdots, ss_{iE_i} = \{c_{ij_{E_i}}, \cdots\}\}$, where each ss_{ij} is a minimum set that may lead to the disclosure. To preserve privacy, PBCP avoids publishing all the contents belonging to any tuple in the sensitive set.

To derive the sensitive sets for a target user, the system assumes adversaries have access to all the contents except for the ones of the target users. The server tests for each user all the combinations of all the user's contents, and discovers the sensitive sets according to the predefined threshold by the user. Then the server records all the minimum sets that may lead to privacy leakage. This procedure may be time-consuming. However, it can be carried out priorly by the server as a preprocessing step. Specifically, the server can maintain the sensitive set each time a user uploads, deletes, or modifies her contents, and notify the user with corresponding changes.

Diverse Vectors and Covered Diverse Vectors. PBCP also maintains diverse vectors for users. A diverse vector refers to a K_0-dimensional vector with binary entries, denoted as $V_i = \{v_{i1}, v_{i2}, \cdots, v_{iK_0}\}$. For each user, all the corresponding entries referring to her diverse contents are set to 1, while all the other entries are 0. For simplicity, we use the terms diverse vector and diverse set interchangeably in this work, both of which indicate the set of diverse contents.

Furthermore, PBCP maintains a covered diverse vector for each user, denoted as $CV_i = \{cv_{i1}, cv_{i2}, \cdots, cv_{iK_0}\}$. Each dimension of the vector has a binary entry indicating whether the corresponding diverse content has been covered. For example, u_k is similar to u_i based on the published contents, and u_k also publishes c_{0j} which is a diverse content for u_i. Then $cv_{ij} = 1$. All the entries for non-diverse contents are always set to 0. PBCP maintains CV_i's to check whether a user is successfully served.

Dependent Graph. PBCP needs a data structure to record the correlations among users, which is the dependent graph. The dependent graph indicates both the similarity and the priorities among users. First of all, users are assigned to different levels according to their privacy preferences. For example, the users with the most flexible preference are at the first level, who have the highest priority. For each pair of users, there is an edge in the dependent graph when these two users are similar. More specifically, consider two users u_i and u_j:

1) When the content publication schemes for both users are not determined yet, or either of the them is not determined, u_i and u_j share an edge if they are similar according to the determined (or published) contents, or there is a feasible publication scheme making u_i and u_j similar.

2) When both u_i and u_j determine their contents for publication, there is no edge between them since they cannot

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2801798, IEEE Transactions on Network Science and Engineering

further contribute to the coverage for each other.

Finally, two users are dependent if they share an edge in the dependent graph. Actually, the dependent graph also provides an approach for integrating the social links among users. For example, two users may also be considered as similar when they are friends or share many common friends. We will consider this part in our future study.

4.3 The Priority-Based Content Publishing Algorithm

The main idea of the algorithm is as follows: PBCP first ranks users according to their priority levels, and initializes the dependent graph to record the correlations among users. Then PBCP processes iteratively from the highest priority level to the lowest one. In each iteration, PBCP utilizes a greedy strategy to publish contents for the users in the current level, and updates the covered diverse sets and the dependent graph. Then PBCP checks whether these users are served. If not, PBCP utilizes another greedy strategy to select the users and contents from lower levels to serve the users in the current level. PBCP mainly has two phases. The first phase initializes the sensitive sets and the diverse vectors. The second phase iteratively determines the set of contents to be published by each user.

Parameters Initialization. In the first phase, PBCP first derives the sensitive sets for each user. For user u_i, PBCP takes all the combinations of u_i's contents as the input, validates them in the inferring function, and derives all the minimum sensitive sets marked as $SS_i = \{ss_{i1}, ss_{i2}, \cdots, ss_{iE_i}\}$, where each ss_{ij} is composed of a set of contents. Next, PBCP initializes the diverse vector for each user, marked as $V_i = \{v_{i1}, v_{i2}, \cdots, v_{iK_0}\}$. $v_{ij} = 1$ means c_{0j} is the diverse content when all the contents are published. Furthermore, PBCP utilizes the covered diverse vector $CV_i = \{cv_{i1}, cv_{i2}, \cdots, cv_{iK_0}\}$ for each user, where cv_{ij} indicates whether c_{0j} is already covered by the published contents of some users similar to u_i. Initially, all cv_{ij}'s are set to 0.

Content Publication. In the second phase, PBCP determines the sets of contents to be published by each user. It considers the privacy issues by avoiding publishing all the contents belonging to any tuple in a sensitive set, maintains a priority level for every user to keep fairness, and chooses the contents to cover the diverse sets for other users. The details of each step are as follows.

1) *Parameter Update:* PBCP first updates the dependent graph according to the current published contents.

2) *Current Level Publication:* PBCP publishes the contents for the current priority level. If PBCP runs for the highest priority, each user u_i in this level locally determines the published contents. PBCP iteratively selects for each user the content appeared in the minimum number of tuples in u_i's sensitive set, until all the contents are published, or publishing any content will lead to the full publication of some tuples.

If PBCP runs for the remaining priority levels, it first checks for each user u_i and each tuple ss_{ij} of u_i in the current level. PBCP searches for all u_i's dependent users u_j's in a higher priority level. When all u_j's, who have a diverse content C_{0k} appearing in ss_{ij}, already have C_{0k} published by other similar users, PBCP removes ss_{ij} from SS_i, and marks content C_{0k} as not publishing. Next, PBCP locally publishes the contents for the users in the current level, which is the same as the first scenario.

3) *Subsequent Level Publication:* After publishing the contents for the users in the current level, PBCP checks whether each user is successfully served, and then publishes the contents for those unserved users. When all the users are served, PBCP goes back to step 1) and processes the next level.

PBCP first updates the covered diverse set for each user and the dependent graph. Furthermore, PBCP sorts the users in the current level according to the reverse order of $\frac{\gamma \sum v_{ij} - \sum cv_{ij}}{DL_i}$, where DL_i is the number of dependent users for u_i in the lower priority levels. In this case, PBCP priorly serves the users who request a smaller number of extra diverse contents, and share more links in the dependent graph. Then for each user u_i in the list, PBCP checks whether u_i achieves the ratio of diverse contents in Equation (3). If yes, PBCP checks for the next user. If no, PBCP selects among the users dependent to u_i for processing. The candidate users are sorted in ascending order of L_{u_i} according to their links in the dependent graph.

For each user u_k in L_{u_i}, if u_k is already similar to u_i according to the published contents, PBCP iteratively selects in C_k the uncovered diverse content for u_i, until no feasible content can be published or C_k does not have any uncovered diverse content for u_i. During the selection, PBCP again selects the contents appearing in the minimum number of tuples. PBCP dynamically checks whether u_i achieves ratio γ, and stops if so. If u_k is not similar to u_i yet, PBCP runs a procedure to derive the similar contents, and then follows the same procedure in the first case. After the procedure in L_{u_i}, PBCP updates the covered diverse vectors and the dependent graph.

Then PBCP checks the next unserved user in the current level, until all the users are considered.

After checking for each user in the current priority level, PBCP goes back to step 1) and considers the users in the next priority level.

PBCP terminates when all the users in all the priority levels have been considered. The pseudo code of PBCP is given in Algorithm 1.

4.4 Algorithm for Selecting Similar Contents

PBCP has a procedure to find similar contents between two users. We propose a greedy strategy to address it. The main idea is to iteratively search for contents appearing in the minimum number of sensitive sets, until two users are similar. Assume the algorithm searches for the contents of u_i to make u_i similar to user u_j in a higher level, and there are still δ' contents remaining to achieve the similarity. The algorithm works as follows:

1) For all the contents in C_i which also appear in C_j, the algorithm ranks them according to their number of appearance times in SS_i.

2) The algorithm iteratively selects the contents from the beginning of the content list, until δ' contents are selected. When the selected contents in the kth round lead to a violation on any tuple in SS_i, the algorithm skips the current content and selects the next one. If none of the contents

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2801798, IEEE Transactions on Network Science and Engineering

Algorithm 1 Priority-Based Content Publishing Algorithm

for Each priority level **do**
  Update the dependent graph via currently published contents;
  for Each u_i in current level **do**
    if Priority level = highest **then**
      Sorting contents according to appearance in SS_i
      while No sensitive sets are violated **do**
        Publishing first content from C_i
      end while
    end if
    if Priority level != highest **then**
      Update SS_i according to previous publication
      Sorting contents according to appearance in SS_i
      while No sensitive sets are violated **do**
        Publishing first content from C_i
      end while
    end if
  end for
  Sorting u_is in current level according to $\frac{\gamma \sum v_{ij} - \sum cv_{ij}}{DL_i}$
  for Each u_i in current level **do**
    if $Q_i < \gamma$ **then**
      Sorting all users in neighbor L_{u_i} according to degree in dependent graph.
      for All users u_k in L_{u_i} **do**
        if $D_{ii'} < \delta$ **then**
          Selecting Similar Contents
        end if
        Sorting all uncovered contents in C_k according to appearance in SS_k
        while $Q_i < \gamma$ **do**
          Publishing first content from C_k
        end while
      end for
    end if
  end for
end for

are qualified, the algorithm traces back, changes the content selected in the $k-1$th round with its next content in the list, and then continues the selection.

5 Performance Analysis

In this section, we first analyze the time complexity of PBCP. Then we prove the fairness and effectiveness of PBCP.

5.1 Time Complexity

In the first phase, the time spent on deriving the sensitive set is $O(N \cdot 2^{K_i} + N \cdot K_0)$, where K_i is the number of contents published by a user, and K_0 is the total number of different contents. As we see, the deriving of sensitive sets can be completed by the server priorly, and multiple techniques can be utilized to accelerate the procedure. Therefore, this part could sometimes be considered as a preprocessing, thus ignored. Furthermore, as each user usually publishes a small set of contents and PBCP only looks for the minimum sensitive set, the actual running time for the first phase could be in the order of $O(N \cdot K_0)$.

In the second phase, the time spent for each iteration is $O(N^2 \cdot K_{max}^2 + K_{max}^2 \cdot |SS| + N^2 \cdot K_{max}^2 \cdot |SS_{max}|)$, which is composed of three parts: updating the dependent graph, publishing contents for the current level, and publishing contents for the subsequent levels. Notice that during the publication in the subsequent levels, PBCP needs to update the dependent graph and search for the similar users. For the first part, it only needs to update the user adjacent to the selected user u_k in L_{u_i}, so the time consumption is $O(D_{dp} \cdot K_{max}^2)$, where D_{dp} is the maximum degree of a user in the dependent graph. For the second part, the time consumption is $O(K_{max}^{\delta})$, which can be solved in polynomial time as δ is a constant. Therefore, the total running time for the second phase is $O(N^2 \cdot K_{max}^2 \cdot |SS_{max}|)$ since there are constant levels of priorities.

Finally, the total running time for PBCP is in the order of $O(N^2 \cdot |SS_{max}| + N \cdot K_0)$, which is determined by the number of users, the number of contents, and the size of a sensitive set.

5.2 Fairness and Effectiveness

PBCP guarantees fairness among users, which means it follows all the principles in Definition 1. We prove this in the following theorem.

Theorem 1. PBCP guarantees fairness among users.

The proof of Theorem. 1 is straightforward. We consider all the cases in our algorithm.

First of all, PBCP guarantees that there are no contents published by high-priority users specifically to cover the diverse contents for low-priority users, or among users with the same priority. Meanwhile, when user u_k with a lower priority does not publish any diverse content for another u_i with a higher priority, one of the following facts is true:

1) u_k is not similar to u_i, which means PBCP does not select u_k.

2) u_k is similar to u_i, but has no contents belonging to the uncovered diverse content set of u_i. Then the service for u_i is free from u_k.

3) u_k is similar to u_i, and has contents belonging to the uncovered diverse content set of u_i. However, publishing any content will violate the sensitive sets. In this case, u_k must be utilized by PBCP in the previous iterations to serve the users with higher priorities than u_i.

In all the remaining cases, PBCP publishes some contents from u_k belonging to the uncovered diverse content set of u_i. Therefore, PBCP follows the principles, where facts 1), 2) and 3) equal to the third, the first, and the second principles in Definition 1. Thus PBCP guarantees fairness.

According to Theorem 1, PBCP will priorly serves users with more flexible preferences on their privacy. This fact indicates that a user more willing to participate in the content publication will receive more refined recommendation results in our framework, achieving a fairness among all participants.

Now we briefly discuss the effectiveness of PBCP. As the existence of priorities could always result in poor performance in terms of global utility, *i.e.*, the total number of successfully served users, it is difficult to evaluate the direct impact on the ratio of served users. However, PBCP still tries to serve more users, while strictly following the fairness

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2801798, IEEE Transactions on Network Science and Engineering

constraint. PBCP tries to avoid the competing publication for multiple higher-priority users, and tries to publish more contents. On the one hand, PBCP always selects the feasible users with the minimum number of links to higher level users in the dependent graph, which means PBCP prefers the users that only need to publish for several other users, and those users serving more higher-level users are postponed temporarily. On the other hand, PBCP always selects the feasible content appearing the minimum number of times in the sensitive set, which means it tries to publish more useful contents from each user. In summary, PBCP can achieve a better performance with more contents published, since it priorly selects the contents useful for either similarity or coverage for others. Then the following theorem indicates the effectiveness in terms of the number of published contents in PBCP.

Theorem 2. The published contents in PBCP are no less than $\frac{1}{D_{dp} \cdot D_{ss}}$ of the global maximum one, where D_{dp} is the maximum degree of a user in the dependent graph, and D_{ss} is the maximum number of appearance times for a content in a sensitive set.

Proof 1. The proof is straightforward. Assume each sensitive set has just two unpublished items, which means one more content can be published from them. In the worst case, for each target user u_j, u_j can locally achieve a $\frac{1}{D_{ss}}$ ratio of published contents compared with the optimal result in step 2) of the second phase. When u_j also utilizes the users in the subsequent levels, u_j may achieve an efficiency ratio $\frac{1}{D_{ss}}$ on each selected user, with totally D_{dp} users. Meanwhile, when the selected user is not similar to u_j, the algorithm for selecting similar contents is utilized. As this algorithm also selects the contents with the minimum number of appearance times in a sensitive set, it can also guarantee a $\frac{1}{D_{ss}}$ ratio on the total number of published contents. Therefore, the overall efficiency ratio is $\frac{1}{D_{dp} \cdot D_{ss}}$.

The degree of a user in the dependent graph could be relatively small when the threshold δ for similarity inceases, and each content generally correlates with a limited size of sensitive information. Therefore, according to Theorem 2, the ratio bound could be small in real-world scenarios.

Finally, PBCP utilizes the existence of fairness to improve effectiveness, which removes some tuples in the sensitive sets during the Current Level Publication. In this case, a target user only needs to serve the users with higher priorities. Therefore, when all the users are processed, the target user can be selfish and only concerns its own publication.

6 Discussion

In this section, we first introduce the applicability of our framework. The composition of sensitive sets are then discussed. Furthermore, different definitions of priority levels are discussed.

6.1 Extensibility of The Framework

Although our framework is designed for the one-time content publication, it could be easily extended for the dynamic scenarios. We can periodically run the algorithm to publish the newly uploaded contents, while preserving both existing and newly emerged sensitive information. They underlying reason is that no sensitive sets will de revealed in our framework. The previous existing sensitive set will still be sensitive, and the newly emerged sensitive set will not be disclosed as the new contents have not been published yet. Meanwhile, our framework can also be extended to online social networks where users may post their ratings in the contents. One simple extension is to consider the common publishing of contents as the same content and similar rating for that content. In this approach, our algorithm can be applied directly to the system, and also distinguishes between the like and the dislike.

6.2 Applicability of The Sensitive Set

Our definition of privacy can be utilized for multiple categories of privacy preservations, ranging from the existence of sensitive contents to the popular differential privacy. For example, when sensitive information is directly included in some contents, a system can utilize our framework to extract all the sensitive contents, and a sensitive set is composed of all such contents. As a second instance, when sensitive information is measurable, like locations, and some contents are close to the sensitive value, a sensitive set is composed of all the contents within a range of the sensitive value, or the combinations of such contents can be used to improve the confidence for the existence of the sensitive value.

Our definition of privacy can also be utilized for the attacks via inferring functions or statistical results, where contents from all users are considered as the input. This principle is achieved by discovering for a target user the combination of contents, the output of which in the inferring functions or statistical results leads to the disclosure of sensitive information. All such combinations form a sensitive set for the target user. Meanwhile, our definition is also similar to differential privacy with the assumption that adversaries know all the contents except for the target user. Finally, our definition of privacy could also be extended for the attacks on binary attributes. In this case, a sensitive set could be simplified to one principle, *i.e.*, balance the number of published contents related to each value of the attribute.

However, our definition of privacy is limited to the number of contents generated by a user. Even though multiple techniques like paralleling or pruning can be utilized and most users only have moderate numbers of contents, it is still time consuming for users with a large number of contents. Therefore, how to efficiently derive a sensitive set is still an unsolved problem. Further study is still needed regarding this issue.

6.3 Priority Levels

There could be numerous definitions for priority levels. Our framework is not limited to any specific definition. Instead, our work could actually take priority level as an input, which mainly determines the assignment of layers in the dependent graph. Therefore, our framework can determine priority according to privacy preference, the willingness in serving other users, the centrality of users, the number of contents owned by users, *etc*.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2801798, IEEE Transactions on Network Science and Engineering

The first two methods consider the subjective attitudes of users, which make data publication more controllable for individuals. However, these methods may lead to poor performance for data publication in some cases, since they are less correlated with the structure of an OSN. Furthermore, users could be unqualified in accurately determining privacy preference, which may further limits the performance.

The third and the fourth methods take the structure of a network into consideration. Therefore, they can achieve better performance on data publication when properly defined. However, priority is in fact passively determined for each user, which is based on a user's role in an OSN. This is sometimes considered to be inappropriate as users are the subjects of data publication.

Generally, it is also an open problem to determine a proper priority level for each user, which balances both user satisfaction and system performance. This is another our future work.

7 Evaluation

7.1 System Settings

We evaluate the performance of the proposed algorithm towards a real world dataset published by Yelp [34], which is an OSN system where users share their experiences on local businesses. More specifically, we focus on the reviewers in Charlotte, NC with more than 30 reviews. For each review, we utilize the first 3 tags in the category of the business to form a set, which serve as the identification of the review. The tags could be "restaurant", "Italian food", "overnight", *etc*.

We also randomly generate sensitive set SS_i for each user. Firstly, the number of tuples in SS_i is achieved by multiplying the number of contents by privacy factor α_j. The various privacy factors refer to different privacy preferences. Generally, a larger α_j indicates more tuples in SS_i, and also a lower priority level for the user.

We compare the proposed algorithm with a baseline local maximum publication mechanism. In this mechanism, each user locally determines her published contents. A user always selects the one that appears the minimum number of times in SS_i within the remaining contents, until publishing any content will lead to a total publication of at least one tuple in SS_i. Two metrics are considered: the ratio of served users and the ratio of published contents. The first metric could directly evaluate the performance of the two algorithms, while the second metric can evaluate the utility of knowledge in an OSN.

7.2 Performance for The Whole Community

In this part, we evaluate the ratio of the successfully served users, as well as the published contents for all users. The evaluation results could validate the general performance of the proposed algorithm for a whole network. All users are randomly assigned with 4 priority levels. The sensitive sets are composed of tuples each including two contents randomly picked from a user, *i.e.*, two contents cannot be published simultaneously. The values of α are 1, 1.5, 2, and 2.5 for each level. We further assume $\delta = 3$, which means two users are considered as similar when they have no fewer than three common contents. This assumption could make a balance between the size of similar users and the recommended results. Finally, we set $\gamma = 0.5$, which is the threshold for successful publication.

In our first group of experiments, we validate the performance with different scales of sensitive sets. Therefore, we introduce a parameter called general privacy preference, denoted as β. Then the sizes of sensitive sets in each level are βK_i, $1.5\beta K_i$, $2\beta K_i$, and $2.5\beta K_i$, where K_i is the number of contents for a target user. The results are shown in Fig.1 and Fig.2

As we can see, PBCP can averagely serve 60% more users compared with the baseline method. This is because when users locally publish their contents, some users cannot be covered since their correlated users do not cooperate. PBCP can properly utilize the correlations to serve users with higher priorities. Furthermore, as the general privacy preference increases from 0.125 to 1.5, both algorithms suffer poor performance, which is due to the fact that constraints from the sensitive sets are over-stringent. In this case, all users tend to be selfish on the publication, and the performance will obviously be limited. Actually, the constraints are still tight even when the general privacy preference is relatively small. The underlying reason is that the existence of sensitive sets limits the total number of published contents. As a consequence, the number of diverse contents for all users is also limited. Meanwhile, we also find in Fig.2 that the number of published contents is comparable with the local maximum method, which indicates our strategy for content selection also achieves local optimization. Therefore, our algorithm will guarantee the publication of each user, while achieving qualified recommended results simultaneously.

The second group of experiments considers the impact of parameter γ, which is the threshold for the successful coverage of a user. As users have different expectations on service quality, this threshold may also vary. The general privacy preference β is 0.5, and γ changes from 0.5 to 0.9. The results are shown in Fig.3 and Fig.4.

As is shown, our algorithm can always outperforms the local maximum method, even if $\gamma = 0.9$. However, the performance for both algorithms degrades significantly as γ increases. The underlying reason is that more contents should be published for each served user, which significantly decreases the number of published diverse contents for the remaining users. This in turns leads to the degradation of the global performance. Furthermore, the published ratio in Fig.4 also partially decreases, which means more contents are published to cover other users, instead of the local maximum number of contents.

The third group of experiment evaluates the performance of PBCP under different values of δ, which means users have different requests for similarity users. The threshold for similarity ranges from 1 to 5, We set $\gamma = 0.5$ and $\beta = 0.5$. The results are shown in Fig.5 and Fig.6.

First of all, the performance of PBCP slightly upgrades as δ increases. The underlying reason is that a larger δ leads to fewer correlations among users. Then a larger number of users do not have diverse contents, and they are always considered to be successful. Meanwhile, the number of published contents in PBCP also approaches that of the baseline algorithm in Fig.5, which means our algorithm

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2801798, IEEE Transactions on Network Science and Engineering

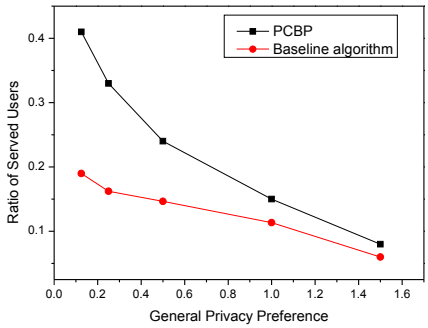


Fig. 1: Ratio of served users under various privacy preferences.

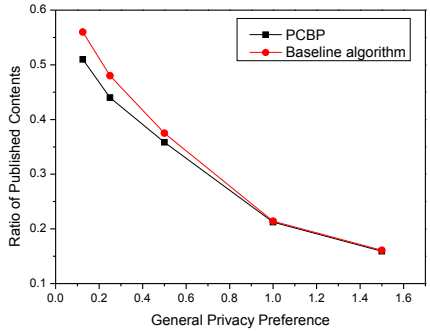


Fig. 2: Ratio of published contents under various privacy preferences.

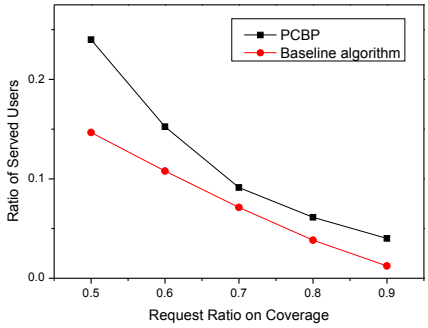


Fig. 3: Ratio of served users under various coverage requests.

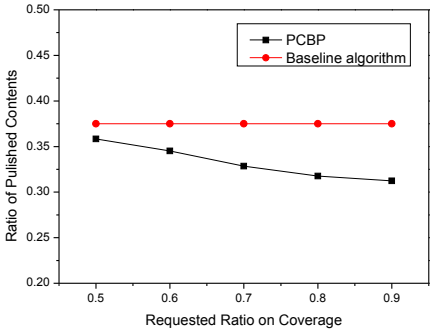


Fig. 4: Ratio of published contents under various coverage requests.

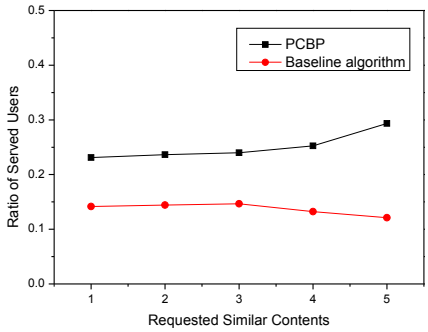


Fig. 5: Ratio of served users under various similarity requests.

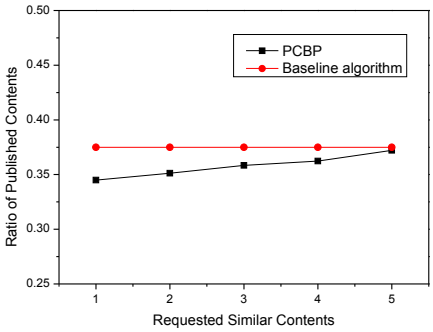


Fig. 6: Ratio of published contents under various similarity requests.

achieves local maximum.

7.3 Performance for Heterogeneous Users

In this part, we investigate the performance of PBCP for different users in OSNs. The evaluation can bring knowledge on the detailed and fine-grained performance of a social network system. Our settings are as follows: general privacy preference $\beta = 0.5$, thresholds for coverage and similarity are $\gamma = 0.5$ and $\delta = 3$, and the remaining settings are the same as the previous group of experiments.

We first evaluate the performance for users with heterogeneous links in the original dependent graph. Generally, the links in the dependent graph refer to the number of similar users. The users are divided into groups according to the numbers of links: [2, 4], [4, 6], [6, 8], [8, 10], [10, $+\infty$]. We investigate in Fig.7 and Fig.8 whether the increasing number of correlated users can lead to a higher coverage.

As shown in Fig.7, the performance remains approximately stable with the increase of the number of links. It is because the number of diverse contents also increases, which is shown in Fig.8. Therefore, the increase on the number of links will not significantly contribute to the replicas of the diverse contents. It is believed that as the number of links keeps increasing, the number of diverse contents will converge, and the performance will improve. However, the considered OSN is sparse and has a large number of contents, which limit the appearance of the convergence.

The second evaluation considers the ratios of successful served users with different priority levels. We can observe from Fig.9 that PBCP serves much more users with higher priority levels, which is twice more than that of the local

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2801798, IEEE Transactions on Network Science and Engineering

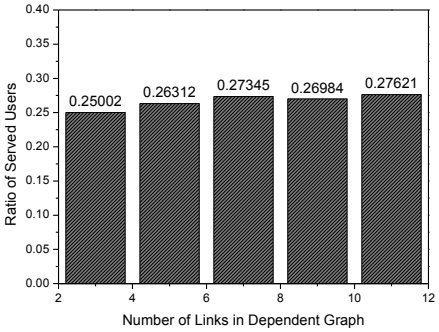


Fig. 7: Ratio of served users under various links.

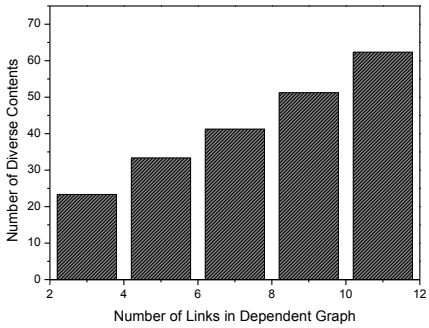


Fig. 8: Number of diverse contents under various links.

maximum method for the first two priory levels. The results indicate that PBCP can guarantee a better performance on fairness among users. Meanwhile, PBCP achieves a comparable performance for lower priority levels. One reason is that when a user publishes and covers diverse contents for high-priority users, a high-priority user may also in return contribute to the coverage for low-priority users.

The last evaluation considers users with different number of contents, and the results are presented in Fig.10 and Fig.11. The users are divided into groups by the number of contents: [30, 39], [40, 49], [50, 59], [60, 69], [70, $+\infty$]. As we can see in Fig.10, the ratio of the served users changes slightly among different groups of users. In this case, although the increase of the number of contents can provide more correlated users, it also causes the increase of the number of diverse contents. As all users in each group are diverse in their priorities, the ratio of the served users will not be significantly changed.

8 Conclusion

As the scale of OSNs keeps growing, the threats and the concerns on sensitive information are more pervasive than ever, which severely thwarts the participation of users as well as limits the promotion of OSN-based services. This paper considers a novel problem where users in OSNs publish their contents for services like content-based recommendation considering both privacy and fairness. A novel framework is proposed, where the privacy concern is presented as a series of sensitive sets, and users are served according to their priority levels. Then a corresponding algorithm is proposed to determine the contents to be published for each user, which also strictly follows the constraints for privacy preservation and guarantees fairness among users. The theoretical analysis is performed to demonstrate the effectiveness of the algorithm, and the evaluations towards a real-world dataset are carried out to validate the performance of the algorithm. We will consider how to integrate the social links into our framework in future work, which is also an essential part of online social networks.

Acknowledgement

This work is partly supported by the National Science Foundation (NSF) under grant NOs. 1252292, 1741277 and 1704287, National Natural Science Foundation of China (NSFC) under Grant NOs. 61502116, 61632010. Foundation of science & technology department of Sichuan province (NO. 2017JZ0031)

References

[1] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 1, pp. 97–107, 2014.

[2] A. Perrin, "Social media usage," *Pew Research Center*, 2015.

[3] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, pp. 1–11, 2018.

[4] T. Georgiou, A. El Abbadi, and X. Yan, "Extracting topics with focused communities for social content recommendation," in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pp. 1432–1443, ACM, 2017.

[5] Y. Wang, G. Yin, Z. Cai, Y. Dong, and H. Dong, "A trust-based probabilistic recommendation model for social networks," *Journal of Network and Computer Applications*, vol. 55, pp. 59–67, 2015.

[6] M. L. Chivers, M. C. Seto, and R. Blanchard, "Gender and sexual orientation differences in sexual response to sexual activities versus gender of actors in sexual films.," *Journal of personality and social psychology*, vol. 93, no. 6, p. 1108, 2007.

[7] T. Pontes, G. Magno, M. Vasconcelos, A. Gupta, J. Almeida, P. Kumaraguru, and V. Almeida, "Beware of what you share: Inferring home location in social networks," in *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*, pp. 571–578, IEEE, 2012.

[8] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy: optimal strategy against localization attacks," in *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 617–627, ACM, 2012.

[9] X. Zheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Follow but no track: privacy preserved profile publishing in cyber-physical social systems," *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 2017.

[10] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," *IEEE Transactions on Information Theory*, vol. 63, no. 6, pp. 4037–4049, 2017.

[11] F. Fei, S. Li, H. Dai, C. Hu, W. Dou, and Q. Ni, "A k-anonymity based schema for location privacy preservation," *IEEE Transactions on Sustainable Computing*, 2017.

[12] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pp. 901–914, ACM, 2013.

[13] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 29–35, 2014.

[14] L. Bonomi, L. Fan, and H. Jin, "An information-theoretic approach to individual sequential data sanitization," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 337–346, ACM, 2016.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2018.2801798, IEEE Transactions on Network Science and Engineering

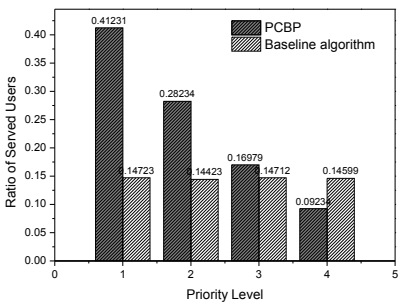


Fig. 9: Ratio of served users with different priorities.

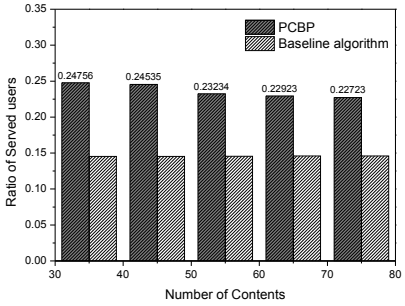


Fig. 10: Ratio of served users under various contents.

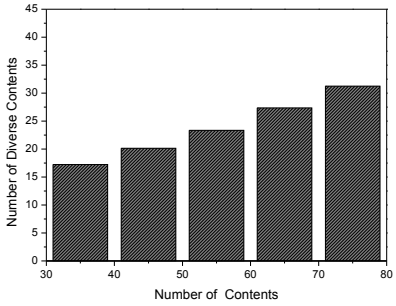


Fig. 11: Number of diverse contents under various contents.

[15] R. Shokri, G. Theodorakopoulos, P. Papadimitratos, E. Kazemi, and J.-P. Hubaux, "Hiding in the mobile crowd: Locationprivacy through collaboration," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 3, pp. 266–279, 2014.

[16] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *The 36th Annual IEEE International Conference on Computer Communications (INFOCOM)*, 2017.

[17] I. Bilogrevic, M. Jadliwala, V. Joneja, K. Kalkan, J.-P. Hubaux, and I. Aad, "Privacy-preserving optimal meeting location determination on mobile devices," *IEEE transactions on information forensics and security*, vol. 9, no. 7, pp. 1141–1156, 2014.

[18] R. I. Ogie, "Adopting incentive mechanisms for large-scale participation in mobile crowdsensing: from literature review to a conceptual framework," *Human-centric Computing and Information Sciences*, vol. 6, no. 1, p. 24, 2016.

[19] Y. Gong, L. Wei, Y. Guo, C. Zhang, and Y. Fang, "Optimal task recommendation for mobile crowdsourcing with privacy control," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 745–756, 2016.

[20] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "Inception: incentivizing privacy-preserving data aggregation for mobile crowd sensing systems.," in *MobiHoc*, pp. 341–350, 2016.

[21] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*, pp. 1–19, Springer, 2008.

[22] E. Shen and T. Yu, "Mining frequent graph patterns with differential privacy," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 545–553, ACM, 2013.

[23] Q. Xiao, R. Chen, and K.-L. Tan, "Differentially private network data release via structural inference," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 911–920, ACM, 2014.

[24] J. Cao and P. Karras, "Publishing microdata with a robust privacy guarantee," *Proceedings of the VLDB Endowment*, vol. 5, no. 11, pp. 1388–1399, 2012.

[25] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proceedings of the forty-second ACM symposium on Theory of computing*, pp. 715–724, ACM, 2010.

[26] E. Shi, H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Annual Network & Distributed System Security Symposium (NDSS)*, Internet Society., 2011.

[27] M. Han, M. Yan, Z. Cai, and Y. Li, "An exploration of broader influence maximization in timeliness networks with opportunistic selection," *Journal of Network and Computer Applications*, vol. 63, pp. 39–49, 2016.

[28] M. Han, M. Yan, Z. Cai, Y. Li, X. Cai, and J. Yu, "Influence maximization by probing partial communities in dynamic online social networks," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 4, 2017.

[29] H. Albinali, M. Han, J. Wang, H. Gao, and Y. Li, "The roles of social network mavens," in *The 12th International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2016)*, pp. 1–12, 2016.

[30] W. Chang, Y.-C. Lin, Y. Lee, and S.-L. Su, "Fairness and safety capacity oriented resource allocation scheme for d2d communications," in *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*, pp. 1–6, IEEE, 2017.

[31] E. Mazzola, M. Piazza, N. Acur, and G. Perrone, "The impact of fairness on the performance of crowdsourcing: An empirical analysis of two intermediate crowdsourcing platforms," 2016.

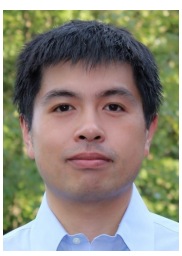
[32] Q. Liu, T. Abdessalem, H. Wu, Z. Yuan, and S. Bressan, "Cost minimization and social fairness for spatial crowdsourcing tasks," in *International Conference on Database Systems for Advanced Applications*, pp. 3–17, Springer, 2016.

[33] R. Faullant, J. Fueller, and K. Hutter, "Fair play: perceived fairness in crowdsourcing communities and its behavioral consequences," in *Academy of Management Proceedings*, vol. 2013, p. 15433, Academy of Management, 2013.

[34] Y. Inc, "Yelp academic dataset." https://www.yelp.com/academic_dataset, 2015.

Xu Zheng received his B.S. and M.S. degree from School of Computer Science and Technology at Harbin Institute of Technology. Mr. Zheng is currently an assistant professor in School of Computer Science and Engineering, University of Electronic Science and Technology of China, and a PhD student in the Department of Computer Science at Georgia State University. Mr. Zheng's research areas focus on wireless network and Big Data.

Guangchun Luo received the Ph.D. degree in computer science from University of Electronic Science and Technology of China, Chengdu, China, in 2004. He is currently a professor and the Associate Dean of computer science at the UESTC. He has published over sixty journal and conference papers in his fields. His research interests include computer networks and big data.



Zhipeng Cai received his PhD and M.S. degree in Department of Computing Science at University of Alberta, and B.S. degree from Department of Computer Science and Engineering at Beijing Institute of Technology. Dr. Cai is currently an Assistant Professor in the Department of Computer Science at Georgia State University. Prior to joining GSU, Dr. Cai was a research faculty in the School of Electrical and Computer Engineering at Georgia Institute of Technology. Dr. Cai's research areas focus on Networking and Big data. Dr. Cai is the recipient of an NSF CAREER Award.