Future Generation Computer Systems 36 (2014) 142–155

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

An effective privacy preserving algorithm for neighborhood-based collaborative filtering

Tianqing Zhu [a,b,*], Yongli Ren [b], Wanlei Zhou [b], Jia Rong [b], Ping Xiong [c]

[a] School of Mathematics and Computer Science, Wuhan Polytechnic University, 68 Xuefu Road, Wuhan 430023, China

[b] School of Information Technology, Deakin University, 221 Burwood Highway, Vic 3125, Australia

[c] School of Information and Security Engineering, Zhongnan University of Economics and Law, 182 Nanhu Road, Wuhan 430073, China

HIGHLIGHTS

- Neighborhood-based Collaborative Filtering faces the privacy issue.
- Private Neighbor Collaborative Filtering provides comprehensive privacy for individuals.
- Recommendation-Aware Sensitivity reduces the magnitude of noise.

ARTICLE INFO

Article history:
Received 15 March 2013
Received in revised form
14 July 2013
Accepted 31 July 2013
Available online 9 August 2013

Keywords:
Privacy preserving
Neighborhood-based collaborative filtering
Differential privacy

ABSTRACT

As a popular technique in recommender systems, Collaborative Filtering (CF) has been the focus of significant attention in recent years, however, its privacy-related issues, especially for the neighborhood-based CF methods, cannot be overlooked. The aim of this study is to address these privacy issues in the context of neighborhood-based CF methods by proposing a *Private Neighbor Collaborative Filtering (PNCF)* algorithm. This algorithm includes two privacy preserving operations: *Private Neighbor Selection* and *Perturbation*. Using the *item-based* method as an example, *Private Neighbor Selection* is constructed on the basis of the notion of *differential privacy*, meaning that neighbors are privately selected for the target item according to its similarities with others. *Recommendation-Aware Sensitivity* and a re-designed *differential privacy* mechanism are introduced in this operation to enhance the performance of recommendations. A *Perturbation* operation then hides the true ratings of selected neighbors by adding Laplace noise. The *PNCF* algorithm reduces the magnitude of the noise introduced from the traditional *differential privacy* mechanism. Moreover, a theoretical analysis is provided to show that the proposed algorithm can resist a *KNN attack* while retaining the accuracy of recommendations. The results from experiments on two real datasets show that the proposed *PNCF* algorithm can obtain a rigid privacy guarantee without high accuracy loss.

1. Introduction

Currently recommender systems are highly successful on e-commerce web sites capable of recommending products users will probably like. *Collaborative Filtering* (CF) is one of the most popular recommendation techniques as it is insensitive to product details. This is achieved by analyzing the user's historical transaction data with various data mining or machine learning techniques, e.g. *k* nearest neighbor rule, the probability theory and matrix factorization [1]. Accordingly, CF methods are generally categorized into the *neighborhood-based* methods and the *model-based* methods [2]. However, there is potential for a breach of privacy in the recommendation process. The literature has shown that continual observation of recommendations with some background information makes it possible to infer the individual's rating or even transaction history, especially for the *neighborhood-based* methods [3]. This is usually referred to as a *KNN attack*, in which an adversary can infer the rating history of an active user by creating fake neighbors based on background information [3]. In this paper, we aim to grapple with the privacy preserving issue in the context of *neighborhood-based* CF methods.

Typically, a collaborative filtering method employs certain traditional privacy preserving approaches, such as cryptographic, obfuscation and perturbation. Among them, *Cryptographic* is suitable for multiple parties but induces extra computational cost [4,5]. *Obfuscation* is easy to understand and implement, but the utility will

* Corresponding author at: School of Information Technology, Deakin University, 221 Burwood Highway, Vic 3125, Australia. Tel.: +61 450790112.
E-mail addresses: tianqing.e.zhu@gmail.com (T. Zhu), yongli@deakin.edu.au (Y. Ren), wanlei@deakin.edu.au (W. Zhou), jiarong@acm.org (J. Rong), pingxiong@znufe.edu.cn (P. Xiong).

0167-739X/$ – see front matter
http://dx.doi.org/10.1016/j.future.2013.07.019

decrease significantly [6,7]. *Perturbation* preserves high privacy levels by adding noise to the original dataset, but the magnitude of noise is subjective and hard to control [8]. Moreover, these traditional approaches suffer from a common weakness: the privacy notion is weak and hard to prove theoretically, thus impairing the credibility of the final result. In order to address these problems, *differential privacy*, a more rigid notion, has been recently proposed [9,10]. Differential privacy provides a strong and provable privacy definition that can quantify the privacy risk to individuals.

Differential privacy was introduced into CF by McSherry et al. [11], who pioneered a study that constructed the private covariance matrix to randomize each user's rating before submitting it to the system. Machanavajjhala et al. [12] presented a graph link-based recommendation algorithm and formalized the trade-off between accuracy and privacy. Both of them employed the *Laplace* noise to mask accurate ratings so the actual opinions of an individual were protected.

Although *differential privacy* is promising for the privacy preserving CF due to its strong privacy guarantee, it still has some limitations and research barriers. More specifically, there are two weaknesses in existing work:

- Existing methods usually fail to hide similar neighbors, which makes CF vulnerable to *KNN attack*. This kind of attack was first mentioned in Calandrino's work [3]. When CF provides similar users or items explicitly or implicitly, the adversary can infer the rating history of a target user by creating fake neighbors based on background information. The *KNN attack* is consequently referred to as a serious privacy violation. Existing privacy methods only protect the rating of the users, but not the users themselves. In actual fact, neighbors can reveal sensitive information about a target user.
- *Differential privacy* usually induces a large noise that affects the quality of the selected neighbors. Existing work usually leads to significant accuracy loss when obtaining sufficient privacy. The large noise occurs for two reasons: the high *sensitivity* and the naive mechanism. Informally, *sensitivity* calibrates the information that needs to be hidden in a query when an individual is deleted in the dataset. It directly determines the size of the noise to be added to each query [9]. Unfortunately, the queries employed in recommendation techniques always have high *sensitivity*, followed by the addition of large noise. Naive mechanism is another issue that leads to high noise. Previous work directly uses the *differential privacy* mechanism and disregards the unique characteristics of recommendations, thus negatively affecting the recommendation performance.

To overcome these weaknesses in *neighborhood-based* CF methods, we propose a *Private Neighbor Collaborative Filtering* (*PNCF*) algorithm in this paper. This idea is based on two observations. Firstly, all possible privacy leakage should be considered. For example, both the ratings and the neighbors are targets that need protection. However, prior work ignores the protection of neighbors. Secondly, *sensitivity* and *mechanism* should be integrated with the requirement of applications. *Differential privacy* was initially proposed as a promising solution to private counting queries [10], whose *sensitivity* is much lower than operations in CF. Hence an adaptive mechanism is expected. The proposed *PNCF* algorithm design is based on these two observations to *provide comprehensive privacy for individuals* while *minimizing the accuracy loss of recommendations*.

To achieve the objective, three issues will be addressed in this paper:

- *How to preserve the neighborhood privacy?* Both a user's neighbors and the original ratings will be hidden in a private CF. How to protect neighbors is the primary issue that needs to be considered. We provide *Private Neighbor Selection* in our algorithm to reduce the probability that an adversary will infer similar users or items from candidates. Independent *Laplace* noise is then added to hide a user's original rating scores. These privacy preserving steps will protect both neighbors and ratings.
- *How to define sensitivity for recommendation purposes?* Traditional *sensitivity* measurement is not suitable for CF due to high dimensional input. How to define a new *sensitivity* is another issue to be addressed. To preserve the performance, we define a practical *Recommendation-Aware Sensitivity* for CF, which reduces the magnitude of noise when compared with the traditional *sensitivity*.
- *How to design the exponential mechanism for CF?* The performance of *neighborhood-based* methods is largely dependent on the quality of selected neighbors. The third issue is how to enhance the quality of selected neighbors in a privacy preserving process. A naive *differentially private* mechanism leads to inferior quality neighbors. Enhancing the quality of neighbors will be a promising way to improve performance. By re-designing the private selection mechanism, we retain the accuracy from the final output result.

The rest of this paper is organized as follows. We present the preliminaries in Section 2, and propose the *PNCF* algorithm in Section 3. In this section, we also undertake theoretical analysis on *sensitivity* in the privacy preserving stage. Section 4 presents results from the experiments, followed by the conclusion in Section 5.

2. Preliminaries

In this section, we introduce the foundational concepts in both *differential privacy* and collaborative filtering, and briefly review the related work.

2.1. Foundational concepts

2.1.1. Notation

Let $U = \{u_1, u_2 \ldots, u_n\}$ be a set of users and $I = \{t_1, t_2 \ldots, t_m\}$ be a set of items. The *user* $\times$ *item* rating dataset R is represented as a $n \times m$ matrix, which can be decomposed into row vectors: $R = [\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_n}]^T$ and $\mathbf{u_a} = [r_{a1}, r_{a2}, \ldots, r_{am}]$. The row vector $\mathbf{u_a}$ corresponds to the user u_a's rating list, and r_{ai} denotes the rating that user u_a gave to item t_i. R can also be represented by column vectors: $R = [\mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_m}]$ and $\mathbf{t_i} = [r_{1i}, r_{2i}, \ldots, r_{ni}]$. For each t_i, $s(i, j)$ represents its similarity with item t_j. $N_k(t_i)$ denotes the set of item t_i's k neighbors, and $U_{ij} = \{u_x \in U | r_{xi} \neq \oslash, r_{xj} \neq \oslash\}$ denotes the set of users, co-rating on both item t_i and t_j. $s(i, j)$ denotes the similarity between t_i with t_j.

2.1.2. Collaborative filtering

Collaborative Filtering (CF), is a well-known recommendation technique and can be further categorized into the *neighborhood-based methods* and the *model-based methods* [13]. The *neighborhood-based* methods are generally based on the k nearest neighbor rule (KNN), and provide recommendations by aggregating the opinions of a user's k nearest neighbors [14].

Two stages are involved in *neighborhood-based methods*: the *Neighbor Selection* and the *Rating Prediction*. In the *Neighbor Selection* stage, the similarity between any two users or any two items are estimated, and correspond to the *user-based* methods and the *item-based* methods. Various measurement metrics have been proposed to compute the similarity. Two of the most popular ones are the *Pearson Correlation Coefficient* (PCC) and *Cosine-based Similarity* (COS) [15]. Neighbors are then selected according to the similarity.

For any item t_i in the *Rating Prediction* Stage, all ratings on t_i by users in $N_k(u_a)$ will be aggregated into the predicted rating $\hat{r}_{ai}$ by