ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200132

Secure Social Recommendation Based on Secret Sharing

Chaochao Chen[1], Liang Li[2], Bingzhe Wu[3], Cheng Hong[4], Li Wang[5], Jun Zhou[6]

Abstract. Nowadays, privacy preserving machine learning has been drawing much attention in both industry and academy. Meanwhile, recommender systems have been extensively adopted by many commercial platforms (e.g. Amazon) and they are mainly built based on user-item interactions. Besides, social platforms (e.g. Facebook) have rich resources of user social information. It is well known that social information, which is rich on social platforms such as Facebook, are useful to build intelligent recommender systems. It is anticipated to combine the social information with the user-item ratings to improve the overall recommendation performance. Most existing recommendation models are built based on the assumptions that the social information are available. However, different platforms are usually reluctant to (or can not) share their data due to certain concerns. In this paper, we first propose a SEcure SOcial RECommendation (SeSoRec) framework which is able to (1) collaboratively mine knowledge from social platform to improve the recommendation performance of the rating platform, and (2) securely keep the raw data of both platforms. We then propose a Secret Sharing based Matrix Multiplication (SSMM) protocol to optimize SeSoRec and prove its correctness and security theoretically. By applying minibatch gradient descent, SeSoRec has linear time complexities in terms of both computation and communication. The comprehensive experimental results on three real-world datasets demonstrate the effectiveness of our proposed SeSoRec and SSMM.

1 Introduction

Nowadays, recommender systems have been extensively used in many commercial platforms [3]. The key point for recommendation is to use as much information as possible to learn better preferences of users and items. To achieve this, besides user-item interaction information, additional information such as social relationship and contextual information have been utilized [19, 29, 2].

Existing researchers usually make the assumption that all kinds of information are available, which is somehow inconsistent with most of the real-world cases. In practice, different kinds of information are located on different platforms, e.g., huge user-item interaction information on Amazon while rich user social information on Facebook. However, different platforms are reluctant to (or can not) share their own data due to competition or regulation reasons.

Therefore, for the recommendation platforms who have rich user-item interaction data, how to use the additional data such as social information on other platforms to further improve recommendation performance, meanwhile protect the raw data security of both platforms, is a crucial question to be answered. It is worthwhile to study such a research topic in both industry and academia.

Secure Multi-Party Computation (MPC) provides a solution to the above question. MPC aims to jointly compute a function for multi-parties while keeping the individual inputs private [35], and it has been adopted by many machine learning algorithms for secure data mining, including decision tree [17], linear regression [27], and logistic regression [25]. However, it has not been applied to the above-mentioned secure multi-party recommendation problems yet.

In this paper, we consider the scenarios where user-item interaction information and user social information are on different platforms, which is quite common in practice. Platform $\mathcal{A}$ has user-item interaction information and Platform $\mathcal{B}$ has user social information, the challenge is to improve the recommendation performance of $\mathcal{A}$ by securely using the user social information on $\mathcal{B}$. To fulfill this, we formalize secure social recommendation as a MPC problem and propose a SEcure SOcial RECommendation (SeSoRec) framework for it. Our proposed SeSoRec is able to (1) collaboratively mine knowledge from social platform to improve the recommendation performance of the rating platform, and (2) keep the raw data of both platforms securely. We further propose a novel Secret Sharing based Matrix Multiplication (SSMM) protocol to optimize SeSoRec, and we also prove its correctness and security. Our proposed SeSoRec and SSMM have linear computation and communication complexities. Experimental results conducted on three real-world datasets demonstrate the effectiveness of our proposed SeSoRec and SSMM.

We summarize our main contributions as follows:

- We observe a secure social recommendation problem in practice, formalize it as a MPC problem, and propose a SeSoRec framework for it.
- We propose a novel Secret Sharing based Matrix Multiplication (SSMM) protocol to optimize SeSoRec, and we also prove its correctness and security.
- Our proposed SeSoRec and SSMM have linear computation and communication complexities.
- Experimental results conducted on three real-world datasets demonstrate the effectiveness of SeSoRec and SSMM.

2 Background

In this section, we review related backgrounds, including (1) social recommendation, (2) secure multi-party computation, and (3) privacy preserving recommendation.

2.1 Social Recommendation

Factorization based recommendation [24, 16, 3, 2] is one of the most popular approaches in recommender system. It factorizes a user-

[1] Ant Financial Services Group, China, email: chaochao.ccc@antfin.com
[2] Huawei Noah's Ark Lab, China, email: liliang103@huawei.com
[3] Peking University, China, email: wubingzhe@pku.edu.cn
[4] Alibaba security, China, email: vince.hc@alibaba-inc.com
[5] Ant Financial Services Group, China, email: raymond.wangl@antfin.com
[6] Ant Financial Services Group, China, email: jun.zhoujun@antfin.com

item rating (or other interaction) matrix into a user latent matrix and an item latent matrix. However, traditional factorization based approaches assume that users are independent and identically distributed, which is inconsistent with the reality that users are inherently connected via various types of social relations such as friendships and trust relations. Therefore, social factorization models incorporate social relationship into account to improve recommendation performance [19], and the basic intuition is that connected users are likely to have similar preferences. According to [32], social factorization models can be formally stated as:
social factorization model = basic factorization model + social information model.

To date, different social information models were proposed to capture social information, and the basic intuition is that connected users are likely to have similar preferences.

2.2 Secure Multi-Party Computation

The concept of secure Multi-Party Computation (MPC) was formally introduced in [34], which aims to generate methods (or protocols) for multi-parties to jointly compute a function (e.g., vector multiplication) over their inputs (e.g., vectors for each party) while keeping those inputs private. MPC can be implemented using different protocols, such as garbled circuits [35], GMW [10], and secret sharing [30]. MPC has been applied into many machine learning algorithms, such as decision tree [17], linear regression [27], logistic regression [25], and collaborative filtering [31]. In this paper, we propose a secret sharing based matrix muliplication algorithm for secure social recommendation.

2.3 User Privacy Preserving Recommendation

Another related research area belongs to privacy preserving recommendation. Recently, user privacy has drawn lots of attention, and how to train models while keeping user privacy becomes a hot research topic, e.g., federated learning and shared machine learning [21, 4]. There are research works adopt garbled circuits to protect user privacy while making recommendation [26]. Some other works use differential privacy to protect user privacy while training recommendation models [22, 14, 23].

Difference between user privacy and data security. User privacy preserving recommendation aims to protect user privacy on the customer side (2C), while data security based recommendation intends to protect the data security of business partners who have already collected users' private data (2B). In this paper, we aim to (1) integrate rating platform and social platform for better recommendation, and (2) protect the data security of both platforms.

3 The Proposed Model

In this section, we first formally describe the secure social recommendation problem, and then present our proposed SEcure SOcial RECommendation (SeSoRec) framework for this problem.

3.1 Problem Definition

Formally, let $\mathcal{A}$ be the user-item interaction platform, and $\mathcal{U}$ and $\mathcal{V}$ be the user and item set on it, with I and J denoting user size and item size, respectively. Let $\mathcal{R}$ be user-item interaction set between user $i \in \mathcal{U}$ and item $j \in \mathcal{V}$, $|\mathcal{R}|$ is the total number of ratings. Let $\mathbf{R}$ be the user-item interaction matrix, with element r_{ij} being the rating of user i on item j. Let $\mathbf{U} \in \mathbb{R}^{K \times I}$ and $\mathbf{V} \in \mathbb{R}^{K \times J}$ denote the user and item latent factor matrices, with their column vectors $\mathbf{u_i}$ and $\mathbf{v_j}$ being the K-dimensional latent factors for user i and item j, respectively. Let $\mathcal{B}$ be the user social platform, and we assume that the social platform $\mathcal{B}$ has the same user set $\mathcal{U}$ as the user-item interaction platform $\mathcal{A}$. We further let $\mathbf{S}$ be the user-user social matrix[7] , with the element s_{if} being the social relationship strength between user i and user f.

The problem of secure social recommendation is that, platforms $\mathcal{A}$ and $\mathcal{B}$ securely keep their own data and model, meanwhile $\mathcal{A}$ can improve its recommendation performance by utilizing the social information of $\mathcal{B}$.

3.2 Secure Social Recommendation Framework

Social recommendation can be formalized as a basic factorization model plus a social information model, based on the assumption that connected users tend to have similar preferences, as described in Section 2.1. Most existing social factorization models have the following objective function

$$\min_{\mathbf{u_i},\mathbf{v_j}} \sum_{i=1}^{I}\sum_{j=1}^{J} f(r_{ij}, \mathbf{u_i}, \mathbf{v_j}) + \gamma \sum_{i=1}^{I}\sum_{f=1}^{I} g(s_{if}, \mathbf{u_i}, \mathbf{u_f}), \tag{1}$$

where $f(r_{ij}, \mathbf{u_i}, \mathbf{v_j})$ is the loss of the *basic factorization model* that restricts the relationship between the true ratings and predicted ratings, $g(s_{if}, \mathbf{u_i}, \mathbf{u_f})$ is the loss of the *social information model* that restricts the preferences of users who have social relations, and γ controls the social restriction strength. A classical example is the Social Regularizer recommendation (Soreg) approach [19], where

$$f(r_{ij}, \mathbf{u_i}, \mathbf{v_j}) = \frac{1}{2} I_{ij} \left(r_{ij} - \mathbf{u_i}^T \mathbf{v_j} \right)^2, \tag{2}$$

$$g(s_{if}, \mathbf{u_i}, \mathbf{u_f}) = \frac{1}{2} s_{if} ||\mathbf{u_i} - \mathbf{u_f}||_F^2, \tag{3}$$

where I_{ij} is the indicator function that equals to 1 if there is an existing user-item interaction pair and 0 otherwise, and $||\cdot||_F^2$ is the Frobenius norm.

Traditional social recommendation frameworks such as Soreg can be efficiently solved by stochastic Gradient Descent (GD). However, the social information model in Equation (3) involves a real number s_{if} which belongs to the social platform $\mathcal{B}$, and two real-valued vectors $\mathbf{u_i}$ and $\mathbf{u_f}$ which are located on the rating platform $\mathcal{A}$, secure computation are not guaranteed due to the breach that $\mathcal{A}$ can easily deduce the values s_{if} belonging to $\mathcal{B}$.

To solve this problem, we propose to use minibatch GD instead of stochastic GD. We use $\mathbf{B}$ to denote the user-item rating set in the current minibatch and $|\mathbf{B}|$ is the batch size. Let $\mathcal{U}_\mathbf{B}$ and $\mathcal{V}_\mathbf{B}$ be the user set and item set in the current batch, $|\mathcal{U}_\mathbf{B}|$ and $|\mathcal{V}_\mathbf{B}|$ be the user and item size. Apparently $|\mathcal{U}_\mathbf{B}| \leq |\mathbf{B}|$ and $|\mathcal{V}_\mathbf{B}| \leq |\mathbf{B}|$. We use $\mathbf{R_B} \in \mathbb{R}^{|\mathcal{U}_\mathbf{B}| \times |\mathcal{V}_\mathbf{B}|}$ to denote the rating matrix in the current batch, $\mathbf{I_B} \in \mathbb{R}^{|\mathcal{U}_\mathbf{B}| \times |\mathcal{V}_\mathbf{B}|}$ to denote the indicator matrix in the current batch. Let $\mathbf{U_B} \in \mathbb{R}^{K \times |\mathcal{U}_\mathbf{B}|}$ and $\mathbf{V_B} \in \mathbb{R}^{K \times |\mathcal{V}_\mathbf{B}|}$ be the latent factors of the corresponding users and items in the current minibatch. Equation (1) becomes

$$\begin{aligned} \min_{\mathbf{U_B},\mathbf{V_B}} \mathcal{L} = & \frac{1}{2} ||\mathbf{I_B} \circ \left(\mathbf{R_B} - \mathbf{U_B}^T \mathbf{V_B} \right)||_F^2 \\ & + \frac{\gamma}{2} \mathbf{SUM} \left(\mathbf{D_B} \circ (\mathbf{U_B}^T \mathbf{U_B}) \right) - \gamma \mathbf{SUM} \left(\mathbf{S_B} \circ (\mathbf{U_B}^T \mathbf{U}) \right) \\ & + \frac{\gamma}{2} \mathbf{SUM} \left(\mathbf{E} \circ (\mathbf{U}^T \mathbf{U}) \right) + \frac{\lambda}{2} \left(||\mathbf{U_B}||_F^2 + ||\mathbf{V_B}||_F^2 \right), \end{aligned} \tag{4}$$

[7] Note that our model can be slightly modifed to meet the case when $\mathbf{S}$ is asymmetric.

Algorithm 1: Secure social recommendation

Input: The observed rating matrix ($\mathbf{R}$) on platform $\mathcal{A}$, user social matrix ($\mathbf{S}$) on platform $\mathcal{B}$, regularization strength(γ, λ), learning rate (θ), and maximum iterations (T)

Output: user latent matrix ($\mathbf{U}$) and item latent matrix ($\mathbf{V}$) on platform $\mathcal{A}$

Platform $\mathcal{A}$ initializes $\mathbf{U}$ and $\mathbf{V}$
for $t = 1$ *to* T **do**
  $\mathcal{A}$ and $\mathcal{B}$ calculate $\mathbf{D}^T\mathbf{U}$ and $\mathbf{S}^T\mathbf{U}$ based on the secure matrix multiplication in Algorithm 2
  $\mathcal{A}$ locally calculates $\frac{\partial \mathcal{L}}{\partial \mathbf{U}}$ based on Equation (5)
  $\mathcal{A}$ locally calculates $\frac{\partial \mathcal{L}}{\partial \mathbf{V}}$ based on Equation (6)
  $\mathcal{A}$ locally updates $\mathbf{U}$ by $\mathbf{U} \leftarrow \mathbf{U} - \theta\frac{\partial \mathcal{L}}{\partial \mathbf{U}}$
  $\mathcal{A}$ locally updates $\mathbf{V}$ by $\mathbf{V} \leftarrow \mathbf{V} - \theta\frac{\partial \mathcal{L}}{\partial \mathbf{V}}$
end
return $\mathbf{U}$ and $\mathbf{V}$ on $\mathcal{A}$

where $\mathbf{D_B} \in \mathbb{R}^{|\mathcal{U}_\mathbf{B}|\times|\mathcal{U}_\mathbf{B}|}$ is a diagonal matrix with diagonal element $d_b = \sum_{f=1}^{I} s_{bf}$, $\mathbf{S_B} \in \mathbb{R}^{|\mathcal{U}_\mathbf{B}|\times I}$ is the social matrix of the users in current minibatch, and $\mathbf{E} \in \mathbb{R}^{I\times I}$ is also a diagonal matrix with diagonal element $e_i = \sum_{b=1}^{|\mathcal{U}_\mathbf{B}|} s_{bi}$. The gradients of $\mathcal{L}$ in Equation (4) with respect to $\mathbf{U_B}$ and $\mathbf{V_B}$ are

$$\frac{\partial \mathcal{L}}{\partial \mathbf{U_B}} = -\mathbf{V_B}\left(\left(\mathbf{R_B} - \mathbf{U_B}^T\mathbf{V_B}\right)^T \circ \mathbf{I_B}\right) + \frac{\gamma}{2}\mathbf{U_B}\mathbf{D_B}^T - \gamma\mathbf{U}\mathbf{S_B}^T + \frac{\gamma}{2}\mathbf{U_B}\mathbf{E_B}^T + \lambda\mathbf{U_B} \tag{5}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{V_B}} = -\mathbf{U_B}\left(\left(\mathbf{R_B} - \mathbf{U_B}^T\mathbf{V_B}\right)^T \circ \mathbf{I_B}\right) + \lambda\mathbf{V_B}, \tag{6}$$

where $\mathbf{E_B} \in \mathbb{R}^{|\mathcal{U}_\mathbf{B}|\times|\mathcal{U}_\mathbf{B}|}$ is a diagonal matrix with diagonal element $e_b = e_{i|i=b}$ which is get by extracting the corresponding users' diagonal elements from $\mathbf{E}$ in current batch.

We observe in Equations (5) and (6) that the matrix product terms $\mathbf{U_B}\mathbf{D_B}^T$, $\mathbf{U}\mathbf{S_B}^T$, and $\mathbf{U_B}\mathbf{E_B}^T$ are crucial. These terms involve one matrix ($\mathbf{U}$ or $\mathbf{U_B}$) on the rating platform and another matrix ($\mathbf{D_B}$, $\mathbf{S_B}$, or $\mathbf{E_B}$) on the social platform. All the other terms can be calculated locally by the rating platform. Therefore, we conclude that *the key to secure social recommendation is the secure matrix multiplication operation,* which is a secure MPC problem. We summarize the proposed SEcure SOcial RECommendation (SeSoRec) solution in Algorithm 1, and will present how to perform secure matrix multiplication in the next section.

4 Secret Sharing based Matrix Multiplication

In this section, we first describe technical preliminaries, and then present a secure matrix multiplication protocol, followed by its correctness and security proof.

4.1 Preliminaries

Secret Sharing. Our proposal relies on Additive Sharing. We briefly review this but refer the reader to [7] for more details. To additively share ($\mathbf{Shr}(\cdot)$) an ℓ-bit value x for two parties ($\mathcal{A}$ and $\mathcal{B}$), party $\mathcal{A}$ generates $x_\mathcal{B} \in \mathbb{Z}_{2^\ell}$ uniformly at random, sends $x_\mathcal{B}$ to to party $\mathcal{B}$, and keeps $x_\mathcal{A} = (x - x_\mathcal{B}) \bmod 2^\ell$. We use $\langle x\rangle_i$ to denote the share of party i. To reconstruct ($\mathbf{Rec}(\cdot,\cdot)$) an additively shared value $\langle x\rangle$, each party i sends $\langle x\rangle_i$ to one who computes $\sum_i x_i \bmod 2^\ell, i \in \{\mathcal{A}, \mathcal{B}\}$. In this paper, we denote additive sharing by $\langle\cdot\rangle$.

Apply to decimal numbers. The above protocol can not work directly with decimal numbers, since it is not possible to sample uniformly in $\mathbb{R}$ [5]. Following the existing work [25], we approximate decimal arithmetics by using fixed-point arithmetics. First, fixed-point addition is trivial. Second, for fixed-point multiplication, we use the following strategy. Suppose a and b are two decimal numbers with at most l_F bits in the fractional part, we first transform them to integers by letting $a' = 2^{l_F}a$ and $b' = 2^{l_F}b$, and then calculate $z = a'b'$. Finally, the last l_F bits of z are truncated so that it has at most l_F bits representing the fractional part. The correctness of the above truncation technique for secret sharing can be found in [25].

Simulation-based Security Proof. To formally prove that a protocol is secure, we adopt the *semi-honest* point of view [9], where each participant truthfully obeys the protocol while being curious about the other parties' original data. Under the *real world* and *ideal world* simulation-based proof [18], whatever can be computed by one party can be simulated given only the messages it receives during the protocol, which implies that each party learns nothing from the protocol execution beyond what they can derive from messages received in the protocol. To formalize our security proof, we need the following notations:

- We use $f(x_1, x_2)$ to denote a function with two variables, where $x_1, x_2 \in \{0,1\}^n$ could be encodings of any mathematical objects, e.g. integers, vectors, matrices, or even functionals. We also use π to denote a two-party protocol for computing f.
- The *view* of the i-th party ($i \in \{\mathcal{A}, \mathcal{B}\}$) during the execution of π is denoted as $\mathbf{view}_i^\pi(x_1, x_2, n)$ which can be expanded as $(x_i, \mathbf{r}^i, \mathbf{m}^i)$, where x_i is the input of i-th party, $\mathbf{r}^i$ is its internal random bits, and $\mathbf{m}^i$ is the messages received or *derived* by the i-th party during the execution of π. Note that $\mathbf{m}^i$ includes all the intermediate messages received, all information derived from the intermediate messages, and also the output of i-th party during the protocol.
- A *probability ensemble* $X = \{X(a,n)\}_{a\in\{0,1\}^*;n\in\mathbb{N}}$ is an infinite sequence of random variables indexed by $a \in \{0,1\}^*$ and $n \in \mathbb{N}$. In the context of secure multiparty computation, a represents each party's input and n represents problem size.

Definition 1 *Two probability ensembles* $P = \{P(a,n)\}_{a\in\{0,1\}^*;n\in\mathbb{N}}$ *and* $Q = \{Q(a,n)\}_{a\in\{0,1\}^*;n\in\mathbb{N}}$ *are said to be computatitionally indistinguishable, denoted by* $P \overset{c}{\equiv} Q$*, if for every non-uniform polynomial-time algorithm* D *and every polynomial* $p(\cdot)$*, there exists an* $N \in \mathbb{N}$ *such that for every* $a \in \{0,1\}^*$ *and every* $n \in \mathbb{N}$*,*

$$|\Pr\{D(P(a,n)) = 1\} - \Pr\{D(Q(a,n)) = 1\}| \le \frac{1}{p(n)}.$$

Definition 2 *Let* $f(x_1, x_2)$ *be a function. We say a two-party protocol* π *computes* f *with information leakage* v_1 *to party* $\mathcal{A}$ *and* v_2 *to party* $\mathcal{B}$ *where each party is viewed as semi-honest adversaries, if there exist probabilistic polynomial-time algorithms* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *such that*

$$\{(\mathcal{S}_1(1^n, x_1, v_1(x_1,x_2)))\}_{x_1,x_2,n} \overset{c}{\equiv} \{(\boldsymbol{view}_1^\pi(x_1,x_2,n))\},$$
$$\{(\mathcal{S}_2(1^n, x_1, v_2(x_1,x_2)))\}_{x_1,x_2,n} \overset{c}{\equiv} \{(\boldsymbol{view}_2^\pi(x_1,x_2,n))\},$$

where $x_1, x_2 \in \{0,1\}^*$ *and* $|x_1| = |x_2| = n$*.*

4.2 Secret Sharing based Matrix Multiplication

Secure matrix multiplication is the key to SeSoRec. There are several approaches for secure matrix multiplication, such as homomorphic encryption [12, 33, 8] and the secret sharing scheme [6], among which secret sharing is much more efficient. Existing secret sharing based matrix multiplication [6] either needs a trusted Initializer (a trusted third party) or expensive cryptographic primitives [15] to generate randomness before computation, i.e., Beavers pre-computed multiplication triplet [1]. We call it Trusted Initializer based Secure Matrix Multiplication (TISMM), which may not be applicable in reality. Besides, TISMM needs to generate many random matrices, causing efficiency concerns.

In this paper, we propose a novel protocol for secure and efficient matrix multiplication using secret sharing. Suppose two parties $\mathcal{A}$ and $\mathcal{B}$ hold matrix $\mathbf{P} \in \mathbb{R}^{x\times y}$ and matrix $\mathbf{Q} \in \mathbb{R}^{y\times z}$ separately, where y is an even number[8]. Our algorithm generalizes the inner product algorithm proposed in [37] to compute the matrix product $\mathbf{PQ}$. We first summarize our proposed Secret Sharing based Matrix Multiplication (SSMM) in Algorithm 2, and then prove its correctness and security.

Algorithm 2: Secret Sharing based Matrix Multiplication (SSMM)

Input: A private matrix $\mathbf{P} \in \mathbb{R}^{x\times y}$ for $\mathcal{A}$, and a private matrix $\mathbf{Q} \in \mathbb{R}^{y\times z}$ for $\mathcal{B}$

Output: A matrix $\mathbf{M} \in \mathbb{R}^{x\times z}$ for $\mathcal{A}$, and a matrix $\mathbf{N} \in \mathbb{R}^{x\times z}$ for $\mathcal{B}$, such that $\mathbf{M}+\mathbf{N}=\mathbf{PQ}$

1 $\mathcal{A}$ and $\mathcal{B}$ locally generate random matrices $\mathbf{P}' \in \mathbb{R}^{x\times y}$ and $\mathbf{Q}' \in \mathbb{R}^{y\times z}$

2 $\mathcal{A}$ locally extracts even columns and odd columns from $\mathbf{P}'$, and get $\mathbf{P}'_e \in \mathbb{R}^{x\times \frac{y}{2}}$ and $\mathbf{P}'_o \in \mathbb{R}^{x\times \frac{y}{2}}$

3 $\mathcal{B}$ locally extracts even rows and odd rows from $\mathbf{Q}'$, and get $\mathbf{Q}'_e \in \mathbb{R}^{\frac{y}{2}\times z}$ and $\mathbf{Q}'_o \in \mathbb{R}^{\frac{y}{2}\times z}$

4 $\mathcal{A}$ computes $\mathbf{P}_1 = \mathbf{P}+\mathbf{P}'$ and $\mathbf{P}_2 = \mathbf{P}'_e + \mathbf{P}'_o$, and sends $\mathbf{P}_1$ and $\mathbf{P}_2$ to $\mathcal{B}$

5 $\mathcal{B}$ computes $\mathbf{Q}_1 = \mathbf{Q}' - \mathbf{Q}$ and $\mathbf{Q}_2 = \mathbf{Q}'_e - \mathbf{Q}'_o$, and sends $\mathbf{Q}_1$ and $\mathbf{Q}_2$ to $\mathcal{A}$

6 $\mathcal{A}$ locally computes $\mathbf{M} = (\mathbf{P}+2\mathbf{P}')\mathbf{Q}_1 + (\mathbf{P}_2 + \mathbf{P}'_o)\mathbf{Q}_2$

7 $\mathcal{B}$ locally computes $\mathbf{N} = \mathbf{P}_1(2\mathbf{Q}-\mathbf{Q}') - \mathbf{P}_2(\mathbf{Q}_2 + \mathbf{Q}'_e)$

8 $\mathcal{B}$ sends $\mathbf{N}$ to $\mathcal{A}$, and $\mathcal{A}$ calculates $\mathbf{M}+\mathbf{N}$

9 **return** $\mathbf{M}+\mathbf{N}$ for $\mathcal{A}$

4.3 Correctness Proof

According to Algorithm 2, we have

$$\begin{aligned}
\mathbf{M}+\mathbf{N} = \quad & (\mathbf{P}+2\mathbf{P}')\mathbf{Q}_1 + (\mathbf{P}_2+\mathbf{P}'_o)\mathbf{Q}_2 \\
& +\mathbf{P}_1(2\mathbf{Q}-\mathbf{Q}') - \mathbf{P}_2(\mathbf{Q}_2+\mathbf{Q}'_e) \\
= \quad & \mathbf{PQ}_1 + 2\mathbf{P}'\mathbf{Q}_1 + \mathbf{P}_2\mathbf{Q}_2 + \mathbf{P}'_o\mathbf{Q}_2 \\
& +2\mathbf{P}_1\mathbf{Q} - \mathbf{P}_1\mathbf{Q}' - \mathbf{P}_2\mathbf{Q}_2 - \mathbf{P}_2\mathbf{Q}'_e \qquad (7) \\
= \quad & \mathbf{PQ}' - \mathbf{PQ} + 2\mathbf{P}'\mathbf{Q}' - 2\mathbf{P}'\mathbf{Q} \\
& +\mathbf{P}'_o\mathbf{Q}'_e - \mathbf{P}'_o\mathbf{Q}'_o + 2\mathbf{PQ} + 2\mathbf{P}'\mathbf{Q} \\
& -\mathbf{PQ}' - \mathbf{P}'\mathbf{Q}' - \mathbf{P}'_e\mathbf{Q}'_e - \mathbf{P}'_o\mathbf{Q}'_e \qquad (8) \\
= \quad & \mathbf{PQ} + \mathbf{P}'\mathbf{Q}' - \mathbf{P}'_o\mathbf{Q}'_o - \mathbf{P}'_e\mathbf{Q}'_e \qquad (9) \\
= \quad & \mathbf{PQ}. \qquad (10)
\end{aligned}$$

Equation (8) is by substituting $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{Q}_1$, and $\mathbf{Q}_2$ in Equation (7) according to Algorithm 2 (Line 4 and Line 5). Equation (9) holds by simplifying Equation (8). The (i,j)-th entry of $\mathbf{P}'\mathbf{Q}'$ is the inner product of the i-th row of $\mathbf{P}'$ and the jth column of $\mathbf{Q}'$. Finally, by matrix definition, the (i,j)-th entry of $\mathbf{P}'_o\mathbf{Q}'_o$ (resp. $\mathbf{P}'_e\mathbf{Q}'_e$) is the inner product of the odd (resp. even) terms in the i-th row of $\mathbf{P}'$ and the odd (resp. even) terms in the j-th column of $\mathbf{Q}'$, so we have $\mathbf{P}'\mathbf{Q}' = \mathbf{P}'_o\mathbf{Q}'_o + \mathbf{P}'_e\mathbf{Q}'_e$ and the last three terms in Line 4 are cancelled. Thus, the correctness is proved.

4.4 Security Proof

Theorem 1 *Protocol of SSMM (Algorithm 2) computes matrix multiplication with information leakage $\mathbf{Q}_e - \mathbf{Q}_o$ to $\mathcal{A}$ and information leakage $\mathbf{P}_e + \mathbf{P}_o$ to $\mathcal{B}$.*

We first give some intuitive discussions on the information disclosure of Algorithm 2. Let $\mathbf{P}_e$ and $\mathbf{P}_o$ be sub-matrices of $\mathbf{P}$ constructed by its even columns and odd columns. Similarly let $\mathbf{Q}_e$ and $\mathbf{Q}_o$ be sub-matrices of $\mathbf{Q}$ constructed by its even and odd rows. As indicated in line 4 of Algorithm 2, $\mathcal{B}$ has $\mathbf{P}_1$, $\mathbf{P}_2$ from $\mathcal{A}$. By extracting the even column sub-matrix $\mathbf{P}^1_e$ and odd column sub-matrix $\mathbf{P}^1_o$ of $\mathbf{P}_1$, $\mathcal{B}$ can calculate $\mathbf{P}_3 = \mathbf{P}^1_e + \mathbf{P}^1_o$. Since $\mathbf{P}_1 = \mathbf{P}+\mathbf{P}'$, we have $\mathbf{P}^1_e = \mathbf{P}_e + \mathbf{P}'_e, \mathbf{P}^1_o = \mathbf{P}_o + \mathbf{P}'_o$. Thus, $\mathcal{B}$ can compute $\mathbf{P}_e + \mathbf{P}_o$ by subtracting $\mathbf{P}_2$ from $\mathbf{P}_3$. Similar arguments will show that $\mathcal{A}$ can compute $\mathbf{Q}_e - \mathbf{Q}_o$ as partial information obtained from $\mathcal{B}$. Although $\mathcal{A}$ and $\mathcal{B}$ both have some level of information disclosed as discussed above, their own private data are still unrevealed.

We then rigorously prove the security level of SSMM, using the preliminary techniques we have given above. Note that we first assume all the matrices are finite field ($\mathbb{Z}_{2^\ell}$), and then apply fixed point decimal arithmetics. Without loss of generality, we first let $\mathcal{A}$ be the adversary and quantify the information leakage to $\mathcal{B}$. The view of $\mathcal{A}$ in real world contains all information of matrices $\mathbf{P}$ and $\mathbf{P}'$ (including their even and odd column sub-matrices), together with $\mathbf{Q}_1$ and $\mathbf{Q}_2$. The key point of the proof is to construct a simulator which can reproduce the same distribution of $\mathbf{Q}_1$ and $\mathbf{Q}_2$. The simulator $\mathcal{S}_A$ for $\mathcal{A}$'s view proceeds like this:

1. Assume $\mathcal{S}_A$ has $\mathbf{Q}_e - \mathbf{Q}_o$ as prior knowledge;
2. $\mathcal{S}_A$ generate random matrix $\mathbf{Q}^\star \in \mathbb{R}^{y\times z}$;
3. $\mathcal{S}_A$ Calculate $\mathbf{Q}^\star_2 = (\mathbf{Q}^\star_e - \mathbf{Q}^\star_o) - (\mathbf{Q}_e - \mathbf{Q}_o)$.

$\mathbf{Q}^\star_e$ and $\mathbf{Q}^\star_o$ are similarly defined as the even column and odd column sub-matrices of $\mathbf{Q}^\star$. We claim that $(\mathbf{Q}^\star, \mathbf{Q}^\star_2)$ has the same distribution as $(\mathbf{Q}_1, \mathbf{Q}_2)$, thus being computationally indistinguishable. To see this, we first notice that $\mathbf{Q}_1$ is the difference between a random matrix $\mathbf{Q}'$ and a fixed matrix $\mathbf{Q}$, which is equally distributed as a random matrix, say $\mathbf{Q}^\star$. With this in mind, it can be seen similarly that $\mathbf{Q}^\star_e - \mathbf{Q}_e$ is equally distributed as $\mathbf{Q}'_e$ and $\mathbf{Q}^\star_o - \mathbf{Q}_o$ is equally distributed as $\mathbf{Q}'_o$. Therefore, $\mathbf{Q}^\star_2$ is equally distributed as $\mathbf{Q}_2$. Moreover, $\mathcal{S}_A$ can reproduce all information of matrices $\mathbf{P}$ and $\mathbf{P}'$. So with additional information of $\mathbf{Q}_e - \mathbf{Q}_o$, the ideal world simulator $\mathcal{S}_A$ successfully reconstructs the view of $\mathcal{A}$, which is equivalent to say that in the real world, only partial information $\mathbf{Q}_e - \mathbf{Q}_o$ has been disclosed to $\mathcal{A}$ after running the protocol.

Similar simulator $\mathcal{S}_B$ can be constructed when assuming $\mathcal{B}$ as the adversary. This completes the security proof.

Complexity Analysis of SSMM. The computational complexity mainly comes from Line 6 and 7 in Algorithm 2, which is $O(x \times y \times z)$. The communication complexity from $\mathcal{A}$ to $\mathcal{B}$ depends on

[8] One can simply change y to an even number by adding an additional zero column in $\mathbf{P}$ and zero row in $\mathbf{Q}$

matrices $\mathbf{P}_1$ and $\mathbf{P}_2$, both of which are $O(x \times y)$. The communication complexity from $\mathcal{B}$ to $\mathcal{A}$ depends on matrices $\mathbf{Q}_1$, $\mathbf{Q}_2$, and $\mathbf{N}$, which are $O(y \times z)$, $O(y \times z)$, and $O(x \times z)$, respectively, and $O((x+y) \times z)$ in total.

When one of the matrices is sparse, we can slightly modify the secret sharing strategy in Algorithm 2 such that both the computational and communication complexities are reduced accordingly. Without loss of generality, we assume $\mathbf{Q}$ is sparse in the sense that for the rows in $\mathbf{Q}$ the average number of non-zero entries is $d \ll z$. When generating $\mathbf{Q}'$, $\mathcal{B}$ does not make it so dense as in Line 1 of Algorithm 2. The new strategy for generating $\mathbf{Q}'$ is as follows:

for *each row in* $\boldsymbol{Q}$ **do**

1. generate random numbers for all non-zero entries
2. randomly select $d' \ll z$ entries from the zero entries and generate random numbers for these entries

end

The value d' is the selected number of non-zero entries of all rows in $\mathbf{Q}$, and the above new strategy makes d' small in order to guarantee that the secret shares from $\mathcal{B}$ to $\mathcal{A}$ are sparse. However, as d' becomes smaller, $\mathcal{A}$ would obtain more information on $\mathbf{Q}$. An extremal case is $d' = 0$, in which $\mathcal{A}$ can infer the overall sparsity of Q. Therefore, a reasonable way is to choose $d' = O(d)$. Note that, in practice, one should keep its strategies of choosing d' (i.e., the ratio of d'/d for each row) privately in case of information leakage. As long as $\mathbf{Q}$ is sparse, $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are both sparse and can be calculated when generating $\mathbf{Q}'$. The computational complexity for matrix multiplication decreases to $O(x \times y \times d)$, and the communication complexity from $\mathcal{B}$ to $\mathcal{A}$ decreases to $O(x \times z + y \times d)$, and thus they are significantly reduced compared to the general case analysis.

We remark that the above new secret sharing strategy for sparse matrix exactly satisfy our requirement in `SeSoRec`. Usually the social matrix $\mathbf{S}$ is sparse. When the user social platform $\mathcal{B}$ shares its secrets, it can use the above new strategy to generate its secret shares. Moreover, the choice of d' can be private to $\mathcal{B}$ only so that the user-item interaction platform $\mathcal{A}$ cannot gain more information based on the shares from $\mathcal{B}$.

5 Analysis

In this section, we analyze the time complexity of `SeSoRec` and discuss its usage and information leakage.

5.1 Complexity Analysis of `SeSoRec`

We first analyze the communication and computation complexities of `SeSoRec`, as shown in Algorithms 1. Recall that I is user number, $|\mathcal{U}_\mathbf{B}|$ and $|\mathcal{V}_\mathbf{B}|$ denote the user and item numbers in the current minibatch respectively, K denotes the dimension of latent factor, and $|\mathcal{R}|$ is the number of ratings (data zise).

Communication Complexity. The communications come from the calculations of $\mathbf{U_B D_B}^T$, $\mathbf{U S_B}^T$, and $\mathbf{U_B E_B}^T$ using `SSMM`. First, for $\mathbf{U_B D_B}^T$ and $\mathbf{U_B E_B}^T$, by refering to the complexity analysis of the modified `SSMM`, their communication costs are both $O(|\mathcal{U}_\mathbf{B}| \times |\mathcal{U}_\mathbf{B}|)$ for each minibatch, and are both $O(|\mathcal{R}|/|\mathbf{B}| \times |\mathcal{U}_\mathbf{B}| \times |\mathcal{U}_\mathbf{B}|) \leq O(|\mathcal{R}| \times |\mathbf{B}|)$ for passing the dataset once. Seconed, for $\mathbf{US}_\mathbf{B}^T$, the communication of $\mathbf{U}$ only needs to be done once for each data pass, and therefore, its communication cost is $O(I \times K)$. To this end, the total communication costs are $O(|\mathcal{R}| \times |\mathbf{B}|) + O(I \times K)$ for passing dataset once. Since, $|\mathbf{B}| \ll |\mathcal{R}|$ and $K \ll I < |\mathcal{R}|$, the total communication cost is linear with data size.

Computation Complexity. Suppose the average number of neighbors for each user on platform $\mathcal{B}$ is $|\mathcal{N}|$. The time complexity of lines 6 and 7 in Algorithm 2 is $O(|\mathcal{U}_\mathbf{B}| \times |\mathcal{N}| \times K)$ for each minibatch, and is $O(|\mathcal{R}|/|\mathbf{B}| \times |\mathcal{U}_\mathbf{B}| \times |\mathcal{N}| \times K) \leq O(|\mathcal{R}| \times |\mathcal{N}| \times K)$ for passing the dataset once. Similarly, the time complexity of the lines 3 and 4 in Algorithm 1 for passing the dataset once is $O(|\mathcal{R}|/|\mathbf{B}| \times |\mathcal{U}_\mathbf{B}| \times |\mathcal{V}_\mathbf{B}| \times K) \leq O(|\mathcal{R}| \times |\mathbf{B}| \times K)$. Since $|\mathcal{N}|, |\mathbf{B}|, K \ll |\mathcal{R}|$, the total computation cost is also linear with data size.

By applying minibatch gradient descent, the communication and computation complexities of `SeSoRec` are both linear with data size and thus can scale to large dataset.

5.2 Discussion

Secure common user identification. Our proposed `SeSoRec` assumes that platforms $\mathcal{A}$ and $\mathcal{B}$ have the same user set in common, so that they can proceed `SSMM`. The essence of secure common user identification is *private set intersection* (PSI). Existing work [28] has provided efficient solution. PSI can be applied to identify common users on two platforms privately before adopting `SeSoRec` in practice, which guarantees that nothing reveals but the IDs of common users.

Information leakage. `SeSoRec` is asymmetric for two parties, that is, the rating platform $\mathcal{A}$ and the social platform $\mathcal{B}$ collaboratively conduct `SSMM` and return the results to $\mathcal{A}$. Therefore, $\mathcal{B}$ reveals more information to $\mathcal{A}$. Although we have proven its security, it may still cause information leakage of $\mathcal{B}$ when $\mathcal{A}$ maliciously initiate `SSMM` iteratively. Suppose $\mathcal{A}$ and $\mathcal{B}$ calculate $\mathbf{PQ}$ using `SSMM`, $\mathcal{A}$ can infer $\mathbf{Q}$ by varying $\mathbf{P}$ and fixing $\mathbf{Q}$ and doing this procedure with enough rounds. A naive solution is to set a constraint on $\mathbf{Q}$ when conducting `SSMM`. As long as $\mathbf{Q}$ (users in each minibatch) is different in each iteration, `SeSoRec` will have no information leakage. We leave better solutions of this as a future work. Moreover, when one matrix is sparse in `SSMM` and the strategies of choosing d' are exposed, the social platform $\mathcal{B}$ may leak some social information to $\mathcal{A}$. Specifically, under this circumstance, the sparsity of the social matrix on $\mathcal{B}$ is leaked to $\mathcal{A}$, however the specific social values are still protected. Therefore, it is crucial that $\mathcal{B}$ keeps its selection of d' for each row of the social matrix privately.

6 Experiments

In this section, we perform experiments to answer the following question. **Q1:** how does `SeSoRec` perform comparing with the classic matrix factorization and unsecure social recommendation models, **Q2:** what is the performance of `SSMM` comparing with the existing TISMM, and **Q3:** how does the social parameter (λ) affect our model performance.

6.1 Setting

We first describe the datasets, metrics, and comparison methods we use in experiments.

Datasets. We use three public real-world datasets, i.e., *Epinions* [20], *FilmTrust* [11] and *Douban* Movie [36]. All these datasets contain user-item ratings and user social (trust) information, and are widely adopted in literature. Note that although rating and social information are both available in these datasets, we realistically assume that they are located on separate platforms without any possibility of data sharing, which has no side-effect on experiments.

Table 1. Dataset statistics. Assuming that rating information exist on $\mathcal{A}$ and social information are available on $\mathcal{B}$.

Dataset	#user	#item	#rating($\mathcal{A}$)	#social($\mathcal{B}$)
Epinions	8,619	5,539	229,920	232,461
FilmTrust	1,508	2,071	35,497	1,853
Douban	13,530	13,363	2,530,594	264,811

Since the original rating matrices of *Epinions* and *Douban* are too sparse, we filter out the users and items whose interactions are less than 20. Table 1 shows the statistics of these datasets after preprocessing, with which we use *five-fold cross validation* method to conduct experiments and evaluate model performance. That is, we split the dataset into five parts, and each time we use four parts as the training set and take the last part as test set.

Metrics. To evaluate model performance, we adopt two types of metrics, Root Mean Square Error (RMSE) and Normalized Discounted Cumulative Gain (NDCG@n), both of which are popularly used to evaluate factorization based recommendation performance in literature [16, 13]. RMSE is defined as

$$RMSE = \sqrt{\frac{1}{|\tau|}\sum_{(i,j)\in\tau}(r_{ij} - \hat{r_{ij}})^2},$$

where $\hat{r_{ij}}$ is the predicted rating of user i on item j, and $|\tau|$ is the number of predictions in the test dataset τ. RMSE evaluates the error between real ratings and predicted ratings, with smaller values indicating better performance. NDCG@n is defined as

$$NDCG@n = Z_n \sum_{n'=1}^{n} \frac{2^{r'_n} - 1}{log_2(n'+1)},$$

where Z_n is a normalizer to ensure that the perfect ranking has value 1 and r'_n is the relevance (real ratings) of item at position n'. NDCG evaluates the ranking performance of recommendation models, with larger values being better. We report NDCG@10 in experiments, and abbreviate it as NDCG.

Comparison methods. Our proposed `SeSoRec` is a novel secure social recommendation model, which is a secure version of Soreg [19]. We compare `SeSoRec` with the following latent factor models:

- **MF** [24] is a classic latent factor model, which only uses the user-item interaction information on platform $\mathcal{A}$. This is the situation where the social platform $\mathcal{B}$ is reluctant to share raw social information with the rating platform $\mathcal{A}$.
- **Soreg** [19] is a classic social recommendation model, which is unsecure in the sense that $\mathcal{A}$ needs the raw data of $\mathcal{B}$.

Note that we do not compare with the state-of-the-art recommendation methods. The reason is: (1) most of them assume the recommendation platform has many different kinds of information such as contextual information [29], which are unfair for our method to compare with, and (2) our focus is to study the difference between traditional unsecure social recommendation models and our proposed secure social recommendation model.

Hyper-parameters. We set the latent factor dimension $K = 10$, batch size $|\mathbf{B}| = 64$, and vary regularizer λ and learning rate θ to choose their best values. We also vary γ in $\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$ to study its effects on `SeSoRec`. For other parameters, e.g., regularizer λ and learning rate θ, we use grid search to find their best values of each model.

Table 2. Performance comparison on three dataset, including RMSE and NDCG.

Dataset	Metrics	MF	Soreg	`SeSoRec`
Epinions	RMSE	1.2687	1.1791	1.1789
	NDCG	0.0363	0.0405	0.0401
FilmTrust	RMSE	1.1907	1.1754	1.1752
	NDCG	0.2042	0.2128	0.2124
Douban	RMSE	0.7489	0.7420	0.7419
	NDCG	0.0749	0.0780	0.0778

Table 3. Running time comparison of `SSMM` and TISMM.

dimension (h)	100	1000	10000
`SSMM`	0.0025	0.3246	40.744
TISMM	0.0060	0.7279	105.83

6.2 Comparison Results (To Q1)

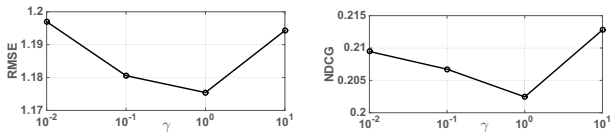
We report the comparison results on three datasets in Table 2. From it, we can observe that: (1) Soreg and `SeSoRec` consistently outperform MF. Moreover, we find that the sparser the dataset is, the more Soreg and `SeSoRec` improve MF. Take RMSE for example, `SeSoRec` improves MF at 7.60%, 1.3%, and 0.98% on *Epinions*, *FilmTrust*, and *Douban*, with their rating densities 0.48%, 1.14%, and 1.4%, respectively. The results prove that social information is indeed important to recommendation performance, especially when data is sparse. (2) Soreg and `SeSoRec` achieve almost the same recommendation accuracy, where the differences come from the fixed point decimal numbers in secret sharing. The result further validates the correctness of our proposed `SSMM` besides the theoretical proof.

6.3 Comparison between `SSMM` and TISMM (To Q2)

As we described in `SSMM` section, existing Trusted Initializer based Secure Matrix Multiplication (TISMM) [6] needs a trusted initializer (a trusted third party) to generate secrets before computation. Although TISMM may not be applicable in practice, we would like to compare the efficiency of our proposed `SSMM` with it. To this end, we randomly generate two square matrices $\mathbf{P} \in \mathbb{R}^{h\times h}$ and $\mathbf{Q} \in \mathbb{R}^{h\times h}$, where h is the dimension of the square matrix. We then report the running time (in seconds) of calculating $\mathbf{PQ}$ using both algorithms in Table 3, where we use local area network. It can be easily seen that our proposed `SSMM` costs much less time than TISMM. The speedup is around 2.4 times on average. This is because TISMM needs to generate more random matrices and involve more matrix operations. Moreover, our proposed `SSMM` protocol does not rely on the trusted initializer which may be difficult to find in practice, thus is more practical.

6.4 Parameter Analysis (To Q3)

Finally, we study the effect of social regularizer parameter γ on `SeSoRec`. Social recommendation can be formalized as a basic factorization model plus a social information model. The social regularizer parameter γ controls the contribution of social information model to the final model performance. The larger γ is, the more likely that the latent factors of connected users are similar, and therefore the more social information model will contribute to the overall performance. Figure 1 shows its effects on *FilmTrust* dataset in terms of



(a) Effect on RMSE (b) Effect on NDCG@10

Figure 1. Effect of γ on *FilmTrust* dataset.

both RMSE and NDCG@10. It can be seen that with a good choice of γ, SeSoRec can balance the contribution of user-item rating data on platform $\mathcal{A}$ and user social data on platform $\mathcal{B}$, and thus, achieve the best performance.

7 Conclusion and Future Work

In this paper, we proposed a secret sharing based secure social recommendation framework, which can not only mine knowledge from social platform to improve the recommendation performance of the rating platform, but also keep the raw data of both platforms securely. Specifically, we first formalized secure social recommendation as a MPC problem and proposed a SEcure SOcial RECommendation (SeSoRec) framework for it. We then proposed a novel Secret Sharing based Matrix Multiplication (SSMM) algorithm to optimize it, and proved its correctness and security. Besides, we analyzed that SeSoRec has linear communication and computation complexities and thus can scale to large datasets. Experimental results on real-world datasets demonstrated that, SeSoRec achieves almost the same accuracy as the existing unsecure social recommendation model, and SSMM significantly outperforms the existing trusted initializer based secure matrix multiplication protocol. In the future, we would like to solve the potential information leakage problem of SeSoRec with better solutions.

REFERENCES

[1] Donald Beaver, 'Efficient multiparty protocols using circuit randomization', in *Cryptology*, pp. 420–432. Springer, (1991).

[2] Chaochao Chen, Kevin Chen-Chuan Chang, Qibing Li, and Xiaolin Zheng, 'Semi-supervised learning meets factorization: Learning to recommend with chain graph model', *TKDD*, **12**(6), 73, (2018).

[3] Chaochao Chen, Ziqi Liu, Peilin Zhao, Longfei Li, Jun Zhou, and Xiaolong Li, 'Distributed collaborative hashing and its applications in ant financial', in *SIGKDD*, pp. 100–109. ACM, (2018).

[4] Chaochao Chen, Ziqi Liu, Peilin Zhao, Jun Zhou, and Xiaolong Li, 'Privacy preserving point-of-interest recommendation using decentralized matrix factorization', in *AAAI*, pp. 257–264, (2018).

[5] Martine de Cock, Rafael Dowsley, Anderson CA Nascimento, and Stacey C Newman, 'Fast, privacy preserving linear regression over distributed datasets based on pre-distributed data', in *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14. ACM, (2015).

[6] Martine De Cock, Rafael Dowsley, Caleb Horst, Raj Katti, Anderson Nascimento, Wing-Sea Poon, and Stacey Truex, 'Efficient and private scoring of decision trees, support vector machines and logistic regression models based on pre-computation', *TDSC*, (2017).

[7] Daniel Demmler, Thomas Schneider, and Michael Zohner, 'Aby-a framework for efficient mixed-protocol secure two-party computation', in *NDSS*, (2015).

[8] Jean-Guillaume Dumas, Pascal Lafourcade, Jean-Baptiste Orfila, and Maxime Puys, 'Private multi-party matrix multiplication and trust computations', *arXiv preprint arXiv:1607.03629*, (2016).

[9] Oded Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*, Cambridge University Press, New York, NY, USA, 2004.

[10] Oded Goldreich, Silvio Micali, and Avi Wigderson, 'How to play any mental game', in *STOC*, pp. 218–229. ACM, (1987).

[11] Guibing Guo, Jie Zhang, and Neil Yorke-Smith, 'A novel bayesian similarity measure for recommender systems', in *IJCAI*, pp. 2619–2625, (2013).

[12] Shuguo Han and Wee Keong Ng, 'Privacy-preserving linear fisher discriminant analysis', in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 136–147. Springer, (2008).

[13] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen, 'Trirank: Review-aware explainable recommendation by modeling aspects', in *CIKM*, pp. 1661–1670. ACM, (2015).

[14] Jingyu Hua, Chang Xia, and Sheng Zhong, 'Differentially private matrix factorization', in *IJCAI*, pp. 1763–1770, (2015).

[15] Marcel Keller, Valerio Pastro, and Dragos Rotaru, 'Overdrive: making spdz great again', in *Eurocrypt*, pp. 158–189. Springer, (2018).

[16] Yehuda Koren, Robert Bell, Chris Volinsky, et al., 'Matrix factorization techniques for recommender systems', *Computer*, **42**(8), 30–37, (2009).

[17] Yehida Lindell, 'Secure multiparty computation for privacy preserving data mining', in *Encyclopedia of Data Warehousing and Mining*, 1005–1009, IGI Global, (2005).

[18] Yehuda Lindell, 'How to simulate it–a tutorial on the simulation proof technique', in *Tutorials on the Foundations of Cryptography*, 277–346, (2017).

[19] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King, 'Recommender systems with social regularization', in *WSDM*, pp. 287–296. ACM, (2011).

[20] Paolo Massa and Paolo Avesani, 'Trust-aware recommender systems', in *Proceedings of the 2007 ACM conference on Recommender systems*, pp. 17–24. ACM, (2007).

[21] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al., 'Communication-efficient learning of deep networks from decentralized data', *arXiv preprint arXiv:1602.05629*, (2016).

[22] Frank McSherry and Ilya Mironov, 'Differentially private recommender systems: Building privacy into the netflix prize contenders', in *SIGKDD*, pp. 627–636. ACM, (2009).

[23] Xuying Meng, Suhang Wang, Kai Shu, Jundong Li, Bo Chen, Huan Liu, and Yujun Zhang, 'Personalized privacy-preserving social recommendation', in *AAAI*, pp. 3796–3803, (2018).

[24] Andriy Mnih and Ruslan Salakhutdinov, 'Probabilistic matrix factorization', in *NIPS*, pp. 1257–1264, (2007).

[25] Payman Mohassel and Yupeng Zhang, 'Secureml: A system for scalable privacy-preserving machine learning', in *S & P*, pp. 19–38. IEEE, (2017).

[26] Valeria Nikolaenko, Stratis Ioannidis, Udi Weinsberg, Marc Joye, Nina Taft, and Dan Boneh, 'Privacy-preserving matrix factorization', in *CCS*, pp. 801–812. ACM, (2013).

[27] Valeria Nikolaenko, Udi Weinsberg, Stratis Ioannidis, Marc Joye, Dan Boneh, and Nina Taft, 'Privacy-preserving ridge regression on hundreds of millions of records', in *S & P*, pp. 334–348, (2013).

[28] Benny Pinkas, Thomas Schneider, and Michael Zohner, 'Faster private set intersection based on {OT} extension', in *USENIX Security*, pp. 797–812, (2014).

[29] Steffen Rendle, 'Factorization machines', in *ICDM*, pp. 995–1000. IEEE, (2010).

[30] Adi Shamir, 'How to share a secret', *Communications of the ACM*, **22**(11), 612–613, (1979).

[31] Erez Shmueli and Tamir Tassa, 'Secure multi-party protocols for item-based collaborative filtering', in *RecSys*, pp. 89–97. ACM, (2017).

[32] Jiliang Tang, Xia Hu, and Huan Liu, 'Social recommendation: a review', *Social Network Analysis and Mining*, **3**(4), 1113–1133, (2013).

[33] Sin G Teo, Vincent Lee, and Shuguo Han, 'A study of efficiency and accuracy of secure multiparty protocol in privacy-preserving data mining', in *WAINA*, pp. 85–90. IEEE, (2012).

[34] Andrew C Yao, 'Protocols for secure computations', in *FOCS*, pp. 160–164, (1982).

[35] Andrew Chi-Chih Yao, 'How to generate and exchange secrets', in *FOCS*, pp. 162–167, (1986).

[36] Erheng Zhong, Wei Fan, Junwei Wang, Lei Xiao, and Yong Li, 'Comsoc: Adaptive transfer of user behaviors over composite social network', in *SIGKDD*, pp. 696–704. ACM, (2012).

[37] Youwen Zhu and Tsuyoshi Takagi, 'Efficient scalar product protocol and its privacy-preserving application', *International Journal of Electronic Security and Digital Forensics*, **7**(1), 1–19, (2015).