Privacy-Preserving Collaborative Recommendation Systems Based on the Scalar Product

Justin Zhan  I-Cheng Wang

Abstract

In the e-commerce era, recommendation systems were introduced to share customer experience and comments. At the same time, there is a need for E-commerce entities to join their recommender system databases to enhance the reliability toward prospective customers and also to maximize the precision of target marketing. However, there will be a privacy disclosure hazard while joining recommender system databases. In order to preserve privacy in merging recommender system databases, we design a novel algorithm based on ElGamal scheme of homomorphic encryption.

Key Words: Privacy, Security, Social Networks.

1 Introduction

A recommender system is a web-based applications that aim at helping customers in the decision making and product selection process. It was introduced to share customer experience and comments in electronic commerce. The most prominent example in e-commerce is the online bookstore amazon.com, where collaborative filtering techniques are used to exploit similarities in the user profile which is based on the navigation and buying history. The main idea is to identify users who presumably have similar preferences and recommend those items which were bought by other users with a similar interest profile. Another technical approach is content-based filtering which builds on the hypothesis that the preferred items of a single user can be extrapolated from her preferences in the past. The third principle approach is to use domain knowledge and to base the recommendations on a thorough understanding of the users current needs, comparable to real-life sales situation. The recommendations are the result of a reasoning process on the domain knowledge which also forms the basis for explaining to the user why an item is proposed. Knowledge-based recommender systems elicit user preferences explicitly, i.e. they provide dynamic personalized and potentially persuasive sales dialogues. Recommender systems are changing from novelties used by a few E-commerce sites, to serious business tools that are re-shaping the world of E-commerce. Many of the largest commerce web sites are already using recommender systems to help their customers find products to purchase.

A recommender system learns from a customer and recommends products that she will find most valuable from all the available products. With many platforms of e-commerce websites, joining recommendation databases can enhance the reliability toward prospective customers. It can also make corporations to maximize the precision of target marketing. Thus, merging the customer-product information between two or more e-commerce entities will be desired. However, due to privacy concerns, different entities may not want to disclose the raw preferences of customers to each other. Furthermore, it may lead to distrust among customers, and the e-commerce entities might lose their markets.

As for the recommender system techniques, there is a detailed taxonomy and examples presented by J. B. Schafer et al. [1] According to that, recommender systems can be categorized into Non-Personalized Recommendations, Attribute-Based Recommendations, Item-to-Item Correlation, and People-to-People Correlation. Each of this taxonomy has a different degree of automation and persistence. In this paper, we focus on the People-to-People Correlation with cosine similarity. From the business point of view, with the advent of E-commerce, the need for personalized services has been emphasized. D. Peppers et al. [9] have advocated the need for one-to-one marketing. As for the privacy issue, J. Canny [10, 11] had proposed some schemes of privacy preserving collaborative filtering. In his schemes, there is a community of user who can compute the aggregation of their private data without disclosing them. Another approach is the randomized perturbation proposed by H. Polat et al. [12, 13]. They deployed a centralized server to store the perturbed numeric ratings, then use these disguised rating to provide the predic-
tions to users. However, their approach cannot achieve 100% accuracy unless all the private data will be disclosed. In this paper, we intend to introduce two approaches which can achieve 100% accuracy while preserving privacy in joining recommendation databases between two entities. We also compare the proposed two approaches and make valuable discussions.

The organization of this paper is as follows. In section 2, we will introduce a recommender systems algorithm. In section 3, we define our problem. In section 4, we propose our solutions to the given problem. In section 5, we present the experimental results. We make a conclusion in section 6.

2 Introduction to Recommender Systems

Recommender systems are a kind of collaborative filtering systems which produce automatic predictions about the interests of a user by collecting preference information from many users. Though there are some issues about the worth of trust on collaborative filtering because of the possibility of biased opinions from person to person. Though providing feedback requires action by the user, less data may be available than with a passive approach, and user expectations may not be met, the advantage of adapting recommender systems to e-commerce is still desired. So as an actual rating given to something of interest by a person who has viewed the topic or product of interest. This produces a reasonable explanation and rank from a reliable source. Another advantage of active filtering is that people want to and ultimately do provide information regarding the matter at hand [8].

Recommender systems form a specific type of information filtering (IF) technique that attempts to present information items (movies, music, books, news, web pages) that are likely to be interested to the user [7]. Recommender systems use product knowledge, e.g., either hand-coded knowledge provided by experts or "mined" knowledge learned from the behavior of consumers, to guide consumers through the often-overwhelming task of locating products like [2]. As we can see the popularity of recommender systems in many e-commerce websites, such as Amazon.com, CDNOW, eBay, Levis, and Moviefinder.com, etc. These recommender systems are intelligent techniques to deal with the problem of information and product overload. They can be utilized to efficiently provide personalized services in most of e-commerce domains, benefiting both the customer and the merchant. Recommender Systems will benefit the customer by making him suggestions on items that he is assumedly like.

In the meanwhile, the business will be benefited by the increase of sales which will normally occur when the customer is presented with more appealing items [3]. There are two basic entities concerned in a recommender system. "User" (also referred to as customer) is a person who utilizes the recommender system to provide his opinion and receive recommendation about items. "Item" (also referred to as product) is being rated by users and the data of them are collected by the system. The inputs of a recommender system are usually arithmetic rating values, which express the opinion of users on items. Ratings are normally provided by the user and follow a specified numerical scale (example: 1-bad to 5-excellent). Ratings can also be gathered implicitly from the user’s purchase history, web logs, hyperlink visits, browsing habits or other types of information access patterns. The outputs of a recommender system can be either predictions or recommendations. A prediction is expressed as a numerical value which represents the anticipated opinion of active user for certain item. This predicted value should necessarily be within the same numerical scale (example: 1-bad to 5-excellent) as the input referring to the opinions provided initially by the active user. This form of recommender systems output is also known as Individual Scoring. On the other hand, a recommendation is expressed as a list of N items, where N ≤ n, which the active user is expected to like most. The usual approach in that case requires this list to include only items that the active user has not already purchased, viewed or rated. This form of recommender systems output is also known as top-N recommendation or ranked scoring. Now we are going to introduce the algorithm of recommender systems. There are three main processes in recommender systems, they are representation, neighborhood formation and recommendation.

2.1 Representation

In the original representation, the input data is defined as a collection of numerical ratings of m users on n items, expressed by the m×n user-item matrix R. We term this user-item matrix of the input data set as original representation. As mentioned earlier, users are not required to provide their opinion on all items. As a result, the user-item matrix is usually sparse, including numerous “no rating” values, making it harder for filtering algorithms to generate satisfactory results. Thus, some techniques, whose purpose is to reduce the sparsity of the initial user-item matrix, have been proposed in order to improve the results of the recommendation process.
2.2 Neighborhood Formation

The core step of the recommendation process is to calculate the similarity between users in the user-item matrix $R$. Users similar to the active user $u_a$ will form a proximity-based neighborhood with him. The active user’s neighborhood should then be utilized in the following step of the recommendation process, in order to estimate his possible preferences. Neighborhood formation has been implemented in two steps: Initially, the similarity between all the users in the user-item matrix, $R$, is calculated with the help of some proximity metrics. The second step is the actual neighborhood generation for the active user, where the similarities of users are processed in order to select those users from whom the neighborhood of the active user consist.

1. Proximity Metrics: The proximity between two users is usually measured using either correlation or cosine measure.

- Pearson Correlation Similarity
  To find the proximity between users $u_i$ and $u_k$, we can utilize the Pearson correlation metric.

  $$sim_{ik} = corr_{ik} = \frac{\sum_{j=1}^{l}(r_{ij} - \bar{r}_i)(r_{kj} - \bar{r}_k)}{\sqrt{\sum_{j=1}^{l}(r_{ij} - \bar{r}_i)^2(\sum_{j=1}^{l}(r_{kj} - \bar{r}_k)^2)}}$$

  It is important to note that the summations over $j$ are calculated over $L$ items for which both users $u_i$ and $u_k$ have expressed their opinions. Obviously, $L \leq n$, where $n$ represents the number of total items in the user-item matrix $R$.

- Cosine Similarity
  In the n-dimensional item space, we can view different users as feature vectors. A user vector consists of $n$ feature slots, one for each available item. The values used to fill those slots can be either the rating $r_{ij}$, that a user $u_i$, provided for the corresponding item, $ij$, or 0, if no such rating exists. Now, we can compute the proximity between two users $u_i$, and $u_k$, by calculating the similarity between their vectors, as the cosine of the angle formed between them.

2. Neighborhood Type

At this point of the recommendation process it is mandatory to distinguish the active user for whom we would like to make predictions, and proceed with generating his neighborhood of users. Neighborhood generation is based on the similarity matrix $S$. There are two neighborhood types, one is center-based, and the other is aggregate.

3. Center-based scheme

Creating the neighborhood by simply selecting from the similarity matrix $S$, and more specifically, from the row of matrix $S$ which corresponds to the active user, those users who have the $L$ highest similarity values with the active user. In other words, it is static.

4. Aggregate neighborhood scheme

Creating the neighborhood by collecting the users who are closest to the centroid of the current neighborhood. The aggregate neighborhood scheme forms a neighborhood of size $L$ by first picking the user who is closest to active user $u_a$. Those two users will now form the current neighborhood, and the selection of the next neighbor will be based on them. In other words, it is a dynamic scheme.

2.3 Recommendation Generation

The final step in the recommendation process is to produce either a prediction, which will be a numerical value representing the predicted opinion of the active user, or a recommendation which will be expressed as a list of the top-$N$ items that the active user will appreciate more. In both cases, the result should be based on the neighborhood of users.

3 Problems

By joining recommender system databases, the e-commerce entities can expand their datasets to provide more precise predictions and recommendations. We assume there are two e-commerce entities, for example on-line bookstores, each of them have similar product sets (ex. books) but the customer sets are not identical. Also, both of them already have their own recommender system with some data records. These two entities want to cooperate with each other to strengthen their recommender system databases and improve the precision of their recommendation towards their own customers. In other words, these entities are going to share their recommendation databases. While sharing the databases with each other, there will be a risk of revealing the raw customer-product information at the same time. This is not a desired result because disclosing raw customer purchase records and preferences will be an infringement against personal privacy. If
customers know their private data are revealed to entities other than the shop they purchased, they will distrust this company, and lead to the collapse of the company. Thus, in order to merge the recommender databases together and not to disclose the actual commercial data at the same time, we have to check the recommender system algorithm to find the vulnerability of potential privacy disclosure. Let us recall the neighborhood formation step in the recommender algorithm. We measure the proximity between two customers by calculating the Pearson correlation similarly as followed. We can assume that, for item (product) $r$, the ratings of $r_i$ and $r_j$ are made by users $u_i$ and $u_k$ from different e-commerce entities. While joining databases together and not to disclose the actual commercial data, we may trust this company, and lead to the collapse of the company. Thus, in order to merge the recommender system algorithm. We measure the proximity between two customers by calculating the Pearson correlation similarity as followed. We can assume that, for item (product) $j$, the ratings of $r_{ij}$ and $r_{kj}$ are made by users $u_i$ and $u_k$ from different e-commerce entities. While joining these two recommender system databases, we have to share the values $(r_{ij} - r_i)$ and $(r_{kj} - r_k)$ with each other. But with value $(r_{ij} - r_i)$, others can get the idea of how much does user $i$ prefer item $j$ compared to item $i$. Now we found the potential infringement of privacy while collaboratively calculating the Pearson correlation similarity for neighborhood in recommender system algorithm.

4 Privacy-Preserving Scalar Product Protocol

The problem of Secure Multiparty Computation (SMC) was first addressed by Yao in his seminal paper [7]. Loosely speaking, SMC deals with the computation of any function with inputs by two or more parties in a distributed network, while ensuring that no more information is revealed than can be inferred from each participant’s input and output. Solutions to the SMC problem have been proposed by Yao [7] and Goldreich et al.[4]. The security of these solutions is based on some cryptographic assumptions, such as the existence of trapdoor permutations. The solutions are generic and elegant, but their prohibitive cost in protecting privacy makes them totally unsuitable for large-scale applications. Therefore, practical solutions need to be developed.

It has been shown that the two-party scalar-product protocol is one of the most important building blocks in the Secure Two-party Computation [4]. On the one hand, the building block of generic circuit evaluation, the oblivious transfer, can be implemented by the scalar-product protocol. On the other hand, it would be much more efficient if we were to construct functions from scalar-products instead of binary gates.

We now introduce the secure two-party product protocol proposed in [4], and prove that it is indeed information-theoretically secure.

Protocol $\pi_2$.

$f_2(x_1,x_2)\mapsto (y_1,y_2)$, where $x_1 \cdot x_2 = y_1 + y_2$.

1. The commodity server generates random numbers $s_1$, $s_2$, and $r_1$, and lets $r_2 = s_1 \cdot s_2 - r_1$. It then sends $(s_1, r_1)$ to $P_1$ and $(s_2, r_2)$ to $P_2$.

2. $P_1$ sends $\hat{x}_1 = x_1 + s_1$ to $P_2$.

3. $P_2$ sends $\hat{x}_2 = x_2 + s_2$ to $P_1$.

4. $P_2$ computes $t = \hat{x}_1 \cdot \hat{x}_2 + r_2 - y_2$ and sends it to $P_1$, where $y_2$ is a randomly generated number.

5. $P_1$ computes $y_1 = t - \hat{x}_2 \cdot s_1 + r_1$.

Theorem 1 (Secure Two-party Product Protocol) $\pi_2$ is an information-theoretically secure protocol that realizes function $f_2$ in the semi-honest adversary model with private channels.

Proof 1 Recall that, according to the problem assumptions, $x_1$, $x_2$, $s_1$, $s_2$, $r_1$, and $y_2$ are independent random variables defined on $GF(p)$; moreover, $s_1$, $s_2$, $r_1$, and $y_2$ are uniformly distributed. From $P_1$’s perspective, in the protocol we have

$$H(x_2|\text{VIEW}^2_{P_1}) = H(x_2|x_1, s_1, r_1, \hat{x}_2, t, y_1). \tag{1}$$

In the protocol, $y_1$ is computed from $s_1$, $r_1$, $\hat{x}_2$, and $t$; therefore, according to Lemma 2, we know that

$$H(x_2|x_1, s_1, r_1, \hat{x}_2, t, y_1) = H(x_2|x_1, r_1, \hat{x}_2, t). \tag{2}$$

Because $t = \hat{x}_1 \cdot \hat{x}_2 + r_2 - y_2$, and $y_2$ is independent of $(x_1, x_2, s_1, s_2, r_1)$, Theorem 2 states that

$$H(x_2|x_1, s_1, r_1, \hat{x}_2, t) = H(x_2|x_1, s_1, r_1, \hat{x}_2). \tag{3}$$

Similarly, because $\hat{x}_2 = x_2 + s_2$, and $s_2$ is uniformly distributed and independent of $(x_1, x_2, s_1, r_1)$, Theorem 2 states that

$$H(x_2|x_1, s_1, r_1, \hat{x}_2) = H(x_2|x_1, s_1, r_1). \tag{4}$$

Then, according to Eqn. (1)~(4) and the independence between $(x_1, s_1, r_1)$ and $x_2$, we have $H(x_2|\text{VIEW}^2_{P_1}) = H(x_2)$.

Next, let us consider $P_2$’s point of view in the protocol:

$$H(x_1|\text{VIEW}^2_{P_2}) = H(x_1|x_2, s_2, \hat{x}_1, r_2, y_2), \tag{5}$$

where $\hat{x}_1 = x_1 + s_1$ and $r_2 = s_1 \cdot s_2 - r_1$. Because $y_2$ is independent of $(x_1, x_2, s_1, s_2, r_1)$, we know that

$$H(x_1|x_2, s_2, \hat{x}_1, r_2, y_2) = H(x_1|x_2, s_2, \hat{x}_1, r_2). \tag{6}$$
Since \( r_2 \) is independent of \((x_1, x_2, s_1, s_2)\) and uniformly distributed, Theorem 3 states that

\[
H(x_1|x_2, s_2, \hat{x}_1) = H(x_1|x_2, s_2).
\] (7)

Again, because of the independence between \(s_1\) and \((x_1, x_2, s_2)\), and by Theorem 3, we have

\[
H(x_1|x_2, s_2, \hat{x}_1) = H(x_1|x_2, s_2).
\] (8)

From Eqn. (5)~(8) and the independence between \(x_1\) and \((x_2, s_2)\) we prove that \(H(x_1|\text{VIEW}^{x_2}_2) = H(x_1)\), which implies that \(\pi_2\) is information-theoretically secure.

Finally, by \(y_1 = t - \hat{x}_2 \cdot s_1 + r_1 = x_1 \cdot x_2 - y_2\), we prove that \(\pi_2\) indeed realizes \(f_2\), which completes the proof.

5 Privacy-Preserving Recommender System

6 Performance Evaluation

In this section, we describe the implementation of our proposed solution. It processes on Intel Pentium M 2.26 GHz processor with 2 GB RAM. We use Microsoft Access 2007 as data storage. And all the programs are written in JAVA, using default APIs. We use a well-known real dataset in this experiment. Jester Joke Recommender System released by Ken Goldberg from UC Berclay. Jester (http://shadow.ieor.berkeley.edu/humor/) has collaborative filtering data of 4.1 Million continuous ratings (-10.00 to +10.00) of 100 jokes from 73,421 users, collected between April 1999 - May 2003. We take the densest sub-dataset of ratings from 23,500 users who have rated 36 or more jokes, a matrix with dimensions 23,500 * 101. And for the representation process of recommendation generation, we add default value 0 for the items not rated. In this early implementation we chose a single active user, then using our algorithm to calculate the similarities between the active 3,000ms for 2048 bits respectively. As we can see, this result presents a exponential cost of time while doubling the encryption key length, that is concordant to the mathematic computation of ElGamal encryption. Second, we implement the algorithm with different size of record involved. We have calculated the similarities in size of 5000, 10000, 15000, 20000, and 23500 records. As it showed in the Figure 3, we can see that both encryption time and transmission time grow linearly while we enlarger the data size.

7 Conclusion and Future Works

Because of the development of E-commerce, recommender systems are more and more popular nowadays. However, customers will not trust the recommendation if the recommender system database is too small. Thus, there is a need for E-commerce entities to join their recommender system databases to enhance the reliability towards prospective customers and also to maximize the precision of target marketing. At the same time we have to preserve the privacy of preferences of customers. With the introduced algorithm, E-commerce entities can merge their recommender system databases without disclosing customer privacy data. For future works, we can apply our homomorphic encryption approach to different taxonomy of recommender system. Thus we might discover some better combinations of techniques for privacy-preserving recommendation systems.