Privacy-Preserving Two-Party Collaborative Filtering on Overlapped Ratings

Burak Memis[1] and Ibrahim Yakut[2]
[1] Dept. of Computer Engineering, Dumlupinar University
Kutahya, - Turkey
[e-mail: burakmemis@dumlupinar.edu.tr]
[2] Dept. of Computer Engineering, Anadolu University
Eskisehir, 26555- Turkey
[e-mail: iyakut@anadolu.edu.tr]
*Corresponding author: Ibrahim Yakut

Received January 23, 2014; revised May 23, 2014; accepted July 7, 2014; published August 29, 2014

Abstract

To promote recommendation services through prediction quality, some privacy-preserving collaborative filtering solutions are proposed to make e-commerce parties collaborate on partitioned data. It is almost probable that two parties hold ratings for the same users and items simultaneously; however, existing two-party privacy-preserving collaborative filtering solutions do not cover such overlaps. Since rating values and rated items are confidential, overlapping ratings make privacy-preservation more challenging. This study examines how to estimate predictions privately based on partitioned data with overlapped entries between two e-commerce companies. We consider both user-based and item-based collaborative filtering approaches and propose novel privacy-preserving collaborative filtering schemes in this sense. We also evaluate our schemes using real movie dataset, and the empirical outcomes show that the parties can promote collaborative services using our schemes.

Keywords: Collaborative filtering, data scarcity, overlapped ratings, Pearson similarity, Slope one predictor, Privacy

A preliminary version of this paper appeared in IEEE WETICE 2013, June 17-20, Hammamet, Tunusia. This version includes additional schemes and analysis for item-based collaborative filtering in extension to the initial work based on user-based collaborative filtering method.

http://dx.doi.org/10.3837/tiis.2014.08.022

1. Introduction

Recommender Systems have recently become very important and popular in the context of e-business applications [1, 2]. Such systems not only facilitate decision process of users having limited time for consuming on the web but also inform internet users about music, film, and books which they intend to taste. *Collaborative filtering* (CF) as a recommender system is useful in the sense that it does not require content analysis for items and provides the ability to recommend items on taste information [3]. User ratings on products are crucial to sustain CF recommendation services. However, collecting and processing user profiles could be a threat to privacy and it is essential to ensure privacy metrics while providing CF services [4, 5]. There are a range of privacy-preserving collaborative filtering (PPCF) schemes dealing with data privacy from collection of user profiles [6, 7] to collaboratively processing sensitive data distributed along a set of data holder parties [8, 9]. In this study, our focus is on PPCF over partitioned data between two parties.

The main goal of CF systems is to conclude customers' preferences with respect to given ratings of similar set of users or items. Hence, there are two different CF approaches with respect to reference entities; these are *user-based* and *item-based* methods. At the launching step, CF is intuitively considered as an user-based phenomenon, and some user-based CF techniques are proposed [3]. In user-based CF methods, user-to-user relations based on similarity and promixity metrics are key elements to drive recommendation mechanisms. Typically, similarities are computed between users, and for each user, neighbor users are determined from the most similar users. Output predictions and recommendations are computed over neighbor users' similarities and ratings. Since Pearson similarity is representative and widely utilized in user-based recommendation algorithms [10], we are going to examine such similarity metrics through our user-based CF investigation.

In order to achieve more accurate CF results in more scalable ways, item-to-item relations are considered, and succeeding CF studies show that item-based *CF* approaches give satisfactory results and even outperform user-based CF in terms of performance and prediction quality [11]. Since item relations are more static than user relations, item similarities can be computed off-line to achieve faster online response with more throughput. Since item-based CF notion introduced by Sarwar et al. [11], many item-based solutions are proposed [12, 13]. In this sense, Lemire and Machlan [12] proposed Slope-one algorithms for recommender systems based on the *popularity differential* intuition. Ratings differences for two item vectors are the key issue to evaluate item-to-item deviations and this makes the method simple but effective to produce predictions. They have been shown to be accurate even with sparse datasets while being updatable on the fly [9]. In this work with respect to such prominent features, we investigate Slope-one predictor that was proposed by Lemire and Mahlachlan [12].

E-commerce companies such as being newly established or expanding product categories suffer from scarcity of ratings. Consequenty, such companies are unable to offer quality CF services. One solution for such problems is to collaborate with another data holder company for featured recommendation services. However, rating data can be subject to privacy risks [4] and e-commerce companies are responsible for the confidentiality of data held by these companies [14, 15]. In order to encourage such parties for cooperation, privacy metrics need to be provided. For this reason, a range of privacy-preserving collaborative filtering (PPCF) schemes are proposed considering partitioned data [14, 16]. By means of privacy-preserving

contribution of bonus data, data scarcity problem can be tackled and companies can provide recommendations having satisfactory quality and quantity.

This study focuses on the following problem: *how can two parties end up with partitioned data having overlapped ratings promote recommendation services ensuring corporate data privacy*? The challenge is to increase the prediction quality while assuring confidentiality of data held by each other. We wish to contribute a study on *overlapped ratings* notion in PPCF on partitioned data and an examination of two-party PPCF solution with overlapped rating data for user-based and item-based CF algorithms. It is also interesting that both algorithms have different proximity metrics; while focused user-based CF is based on Pearson similarities, Slope-one runs on item-to-item deviations. Additionally, we can comprehensively offer solutions for two distinct approaches on overlapped ratings and observe the effects of overlaps in terms of both classes.

The paper is organized, as follows: in the next section, we highlight the significance of our study through the related literature while introducing preliminaries and research problem in the section 3. We are going to demonstrate privacy-preserving user-based and item-based CF solutions in section 4. After theoretically analyzing our proposals in the beginning of section 5, we present experimental setup and results in the the same section. Finally, we conclude the study and give future research directions.

2. Related Work

Cooperative data mining over partitioned data is widely offered for data scarcity problems, and two parties can end up with three kinds of data partitioning settings: *horizontal*, *vertical*, or *arbitrary* [17]. Polat and Du [16] offer top-*N* recommender solution operating on horizontally partitioned data belonging to disjoint set of users for the same items. The same authors [18] introduce PPCF problem over vertically partitioned data where there are ratings for disjoint set of items from common set of users. Kaleli and Polat [19] examined binary predictions on like and dislike values of users between two parties via naïve Bayesian classifier in a privacy-preserving manner over horizontally and vertically partitioned data. Yakut and Polat [20] considered how to provide recommendations using singular value decomposition over horizontally and partitioned data. Hsieh et al. [21] exploited an El Gamal-based homomorphic encryption to join two parties' ratings data. Zhan et al. [22] proposed two-party PPCF approaches with commodity server and compared their schemes with the one proposed in [21].

There have been PPCF studies examining how data is distributed among more than two parties. In this context, PPCF mechanism based on self-organizing map is proposed for vertically distributed data along multi-parties [23]. The same authors [24] examined how to provide recommendations using rating-derived trust metrics on vertically distributed data with privacy. Rather than investigating symmetrically behaving parties, Zhao et al. [25] introduced shared collaborative filtering approach in which parties have asymmetric roles, i.e., while contributor party's data improves the beneficiary party's CF performance, privacy of contributed data cannot be compromised. In all work examining horizontal and vertical partitioned data, no overlaps are expected since authors concentrate on perfectly disjoint set of users or items. Bilge et al. [26] reviewed the state-of-art techniques, from the viewpoint of privacy basics of PPCF, and recently developed mechanisms with the emphasis on the partitioning in data.

In contrast to horizontally and vertically partitioned data, the most remarkable case is the arbitrary partition which consists of arbitrary entries for the same set of users and items in any

party [27]. Yakut and Polat [8] examine how two parties can provide recommendations using item-based CF techniques on arbitrarily partitioned data. Moreover, the same authors [14] propose a two-party CF scheme providing binary predictions via naïve Bayesian classifier over arbitrarily partitioned data with privacy. In another PPCF study, arbitrarily partitioned data is evaluated with other partitioning cases [9]. In the aforementioned study, the authors examined the Slope-one predictor; however, like the other PPCF studies over arbitrarily partitioned data, the schemes are proposed based on the assumption that all the ratings uniquely exist among the parties. Hence, rating overlaps are out of the scope of the work [9]. This study focuses on arbitrarily partitioned data. Additionally, we examine the additional issue of overlapped ratings. It is worth to study in context of PPCF, as typical user-item ratings data is very sparse and rated items may be as sensitive as rating values from the privacy perspective. Such overlaps increase the complexity of problems regarding how to tackle privacy-preservation and the effect on the prediction quality of recommender system. In this sense, we are going to offer solutions with and without handling overlapping ratings; additionally, we reflect on the effects of overlapping ratings on prediction quality.

3. Preliminaries

3.1 User-based Collaborative Filtering with Pearson Similarity

One main task of CF systems is to produce a prediction p_{aq} for an *active user* (a), about *the target item* (q) using $n \times m$ user-item rating matrix where n and m are the number of users and items, respectively. There are mainly three steps in a typical CF process: similarity computations, neighborhood determination, and prediction generation based on the similarity weighted average of neighbor's ratings on q. According to Herlocker et al. [3], similarity between a and train user u can be computed using Pearson correlation coefficient:

$$w_{au} = \frac{\sum_{j \in C} (r_{aj} - \bar{r}_a) \times (r_{uj} - \bar{r}_u)}{\sigma_a \times \sigma_u} \tag{1}$$

where C, w_{au}, r_{uj}, $\bar{r}_u$ and σ_u represent commonly rated items, similarity between a and train user u, the given rating value by u on item j, user u's mean and user u's standard deviation, respectively [2]. After calculating similarity between a and each train user u, a's neighborhood is determined from the best similar users. Then, the final prediction p_{aq} equals to the similarity weighted average of ratings given by the neighbors for q:

$$p_{aq} = \bar{r}_a + \frac{\sum_{u \in N} w_{au} \times (r_{uq} - \bar{r}_u)}{\sum_{u \in N} w_{au}} \tag{2}$$

where, N stands for a's neighbors [3].

3.2 Item-based Collaborative Filtering with Slope-one Predictor

Slope-one predictor algorithms [12] evaluate how much an item is likely to be compared to another one using predictors of the form $f(x) = x + b$. One way to measure this differential is by simply subtracting the average rating of the two items. Deviation dev_{jk} between items j and k can be computed by the following:

$$dev_{jk} = \frac{\sum_i (r_{ij} - r_{ik})}{card_{jk}} \tag{3}$$

where $card_{jk}$ is the cardinality of the set of users i who have rated both items j and k. In order to take the number of ratings observed into consideration, a weighted Slope-one prediction formula is introduced in [12]. Hence, prediction p_{aq} can be computed through the following:

$$p_{aq} = \frac{\sum_j (dev_{qj} + r_{aj}) \times card_{qj}}{\sum_j card_{qj}} \tag{4}$$

where j is each of the available items except q.

3.3 Arbitrarily Partitioning and Overlapped Ratings

Two parties, say A and B, want to provide CF services on partitioned data with overlapped ratings. They have similar sets of customer and item portfolios. According to **Fig. 1**, with respect to rating belongings there are three subsets of ratings: R_A, R_B and R_ϕ. While R_A and R_B hold ratings only belong to A and B, respectively, R_ϕ includes overlapped ratings given by the same user for the same item to the both parties. If R_ϕ is empty, there is no rating overlap and the partitioning case becomes arbitrarily partitioned data (APD) as examined in [8]. However, as discussed in section 2, such overlaps make our study more challenging through prediction quality and privacy-preservation compared to APD. **Fig. 1** also demonstrates the sparsity of CF rating data which have many unrated items shown with empty cells. In our configuration, for the sake of simplicity, we assume that overlapped ratings are consistent, thus, users have already given the same rating value for the same item in both parties' data.

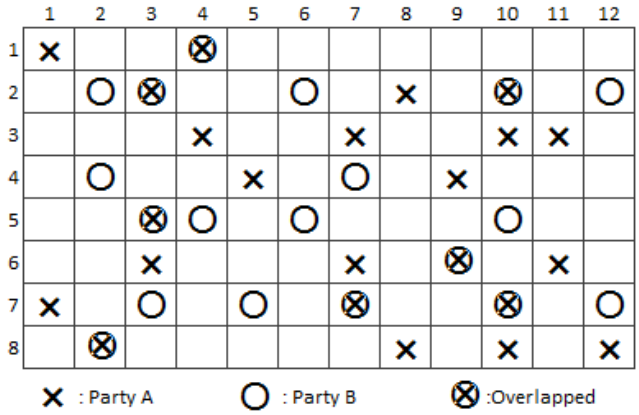

Fig. 1. Partitioned data with sample overlapped ratings

3.4 Privacy Problem

In the context of PPCF [14], *the private* denotes each rating values and also denotes which items are rated by which user. To achieve privacy-preservation, there should be no direct exchange of each individual rating values and rated items without sharing any intermediate

and aggregate values that may reveal individual private information. This is necessary as parties are *semi-honest* and greedy about gathering as much private data as possible, while obeying the predefined procedure. Note that there is no problem for parties to learn which ratings are overlapped, and the information about which ratings are overlapped can be considered public information. Since the value of overlapped ratings for the same user-item pair is equal, we consider that any party's awareness of whether such overlapped item is rated to be a nonissue regarding privacy. After introducing all the related preliminaries, the concentrated *problem* can be described to be in the junction of two viewpoints:

i. From the prediction quality viewpoint, proposed schemes should promote user-based and item-based CF services of the two parties suffering from data scarcity.
ii. From the privacy viewpoint, privacy is preserved when proposed protocols executed by semi-honest parties ending up with arbitrarily partitioned data with overlaps.

In order to solve this problem, the proposed solutions should cater to both the aforementioned viewpoints. Since efficiency is the conflicting goal with respect to prediction quality and privacy-preservation, the solution should promise agreeable computational performance as well.

4. Privacy-Preserving Collaborative Filtering on Overlapped Ratings

In this section, we examine how to perform privacy-preserving of user-based and item-based CF over arbitrariry partitioned data with overlapping ratings. To achieve privacy-preservation through our schemes, we are going to exploit default votes and homomorphic cryptosystems (HCs). Based on public cryptosystems infrastructures, Paillier HC [28] can perform addition of two numbers as ciphertext and obtain encrypted version of the actual sum. Suppose that a and b are two numbers and ξ_K is encryption function with key (K). Then, the ciphertexts of the numbers are $\xi_K(a)$ and $\xi_K(b)$ and their multiplication is $\xi_K(a) \times \xi_K(b) = \xi_K(a + b)$. Additionally in an analogous manner multiplication of plaintext can be performed as $\xi_K(a)^b = \xi_K(ab)$. Paillier HC has self-blinding property permitting public modification of ciphertexts by multiplying with R^N without affecting the plaintext, where R is a random integer value and N is modulus of the operated public cryptosystem. In the following subsections, we will introduce our schemes in detail.

4.1 Privacy-Preserving User-based CF on Overlapped Ratings

Alternatively, we give two-fold solution framework for privacy-preserving user-based CF as seen in **Fig. 2** where building boxes are from any party j's side while k stands for the other party.In our first solution, namely the plain scheme, we investigate the problem without eliminating overlaps while the ultimate scheme determines overlaps privately and then eleminates them.

4.1.1 Preprocessing

Regarding Eq. 1, it can be said that each party needs to normalize its own data. To perform such normalization, each party needs user means. In order to determine the denominator in the same equation they need the standard deviation of each user. Mean and standard deviation are statistically algebraic measures which are composed of distributed measures. Distributed measures can be easily calculated in distributed manner. For example, arithmetic mean equals *sum* of numbers in an array divided by the *count* of this array. If the array is

partitioned among the two parties then by exchanging partial sum and partial size each party can obtain mean of the elements in the array. However for our study, direct exchange of such statistical measures may cause some privacy breaches especially if there are a small amount of available ratings from a user.

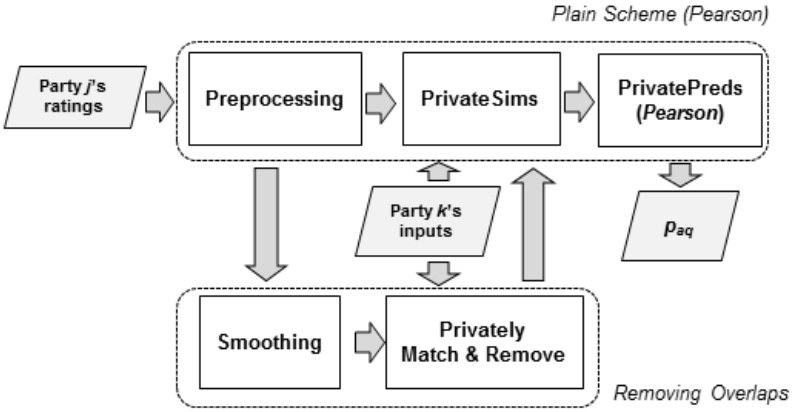

Fig. 2. Privacy-preserving user-based CF on overlapped ratings

To ensure privacy, we offer randomized default vote filling procedure where default votes can be row mean, column mean, or overall mean from available ratings of a party P. After parties agree on *level of filling* (θ) in percentage of density, party P can enhance its own data with v_ds as given below:

1. Randomly or selectively determine β_P from the range [0, θ].
2. Randomly select $\beta_P \cdot \delta_P\%$ of unrated cells where δ_P is the number of available ratings.
3. Fill such selected cells with v_ds.

After filling its own data, parties can exchange partial sum and count values and estimate user mean. Then, they normalize their data using deviation from user mean approach and estimate user standard deviation similar to mean estimation. After preprocessing, each party ends up with estimates of user mean and standard deviation.

4.1.2 Similarity Computation

To compute similarities, two complete user profiles are needed. However, such profiles are arbitrarily distributed among two parties. Hence, there are two parties and two users then the similarity between users a and u can be considered as follows:

$$w_{au} = XY = X_A Y_A + X_A Y_B + X_B Y_A + X_B Y_B \tag{5}$$

where X and Y represent the normalized rating profiles of a and u, respectively; X_P and Y_P stand for available part of such profiles in party P. Overlaps affect the accuracy of the recommender, however, we can hypothesize that explainable results can be obtained despite of overlapped ratings. In plain approach, we give private similarity computation protocol (*PrivateSims*), which does not consider overlaps. Moreover, we also provide how to tackle with overlaps with preserving privacy in the following subsections.

PrivateSims: Private similarity computation protocol

For each user with a the following is performed:

1 Each party assigns zero to all unrated cells.
2 Each party P computes $X_P Y_P$.
3 For train user u being 1 to $n/2$
  3.1 A encrypts each element i of X_A and Y_A with its public key KA
  3.2 A sends all $\xi_{KA}(X_{Ai})$ and $\xi_{KA}(Y_{Ai})$ to B.
  3.3 B computes all $\xi_{KA}(X_{Ai})^{Y_{Bi}}$ then finds $\xi_{KA}(X_A Y_B)$.
  3.4 B computes all $\xi_{KA}(Y_{Ai})^{X_{Bi}}$ then finds $\xi_{KA}(X_B Y_A)$.
  3.5 B encrypts $X_B Y_B$ with KA.
  3.6 Using Paillier's addition, B finds $\xi_{KA}(X_A Y_B + X_B Y_A + X_B Y_B)$.
  3.7 B sends resultant ciphertext to A.
  3.8 A decrypts it, adds $X_A Y_A$ to it and divide proper $\sigma_a \cdot \sigma_u$ and obtains w_{au}.
4 For the remaining train users
  4.1 By switching roles, repeat steps 3.1–3.8.
5 Finally, each party has $n/2$ pieces of n similarities.

PrivateSims protocol's privacy mechanism is based on Paillier HC. In the initial step, we set unrated cells to zero since we intend to utilize absorbing element property of zero during multiplication. In step 2, each party performs partial similarity calculation over only available ratings. With steps 3–4, each party privately computes components of w_{au} and end up with half of the total similarity values between a and each train user u. Note that we exploit self-blinding property of Pailler HC for all encryptions in our scheme in order to discriminate similar plaintexts from each other.

4.1.3. Prediction Computation

Now, we need to compute Eq. 2. Considering that similarities and ratings are distributed among the parties, Eq. 2 can be rearranged as follows:

$$p_{aq} = \bar{r}_a + \frac{\sum_{u \in N} (w_{auA} \times \tilde{r}_{uqA} + w_{auA} \times \tilde{r}_{uqB} + w_{auB} \times \tilde{r}_{uqA} + w_{auB} \times \tilde{r}_{uqB})}{\sum_{u \in N} (w_{auA} + w_{auB})} \tag{6}$$

where w_{auP} and $\tilde{r}_{uqP}$ stands for similarity values and normalized rating of u on q held by party P, respectively. We propose private prediction computation protocol (PrivatePreds) for distributed Pearson similarities and ratings. First of all, such protocol is demonstrated for the case where A is master party (MP) queried for p_{aq}. If MP is B then they must switch the roles and move further. We determine a's neighbors based on threshold (τ) and select neighbors comprised of users having similarities greater than τ in step 1. In step 3, each party generates binary clone rating vector whose entries having value of one if q is rated by u otherwise it is zero. Since one is an identity element for multiplication, we use the binary clones to add up proper similarity values in the denominator. In steps 4–8, B computes for the numerator while in step 9 computations are performed for the denominator. In step 11, $(w_{auA})_P$ stands for similarity values available in A exploited in numerator calculation by party P. At the end of PrivatePreds, MP returns prediction p_{aq} to a.

PrivatePreds (*Pearson*): Privately prediction computation protocol for Pearson similarity

1 Each party assigns zero to all its similarity values less than τ.
2 Each party assigns zero to all unrated cells for q.
3 Each party P generates binary clone rating vector (c_{uqP}).
4 A encrypts each element i of w_{auA}, $\tilde{r}_{uqA}$, and c_{uqA} with KA
5 A sends all $\xi_{KA}(w_{auA})$ and $\xi_{KA}(\tilde{r}_{uqA})$ values to B.
6 B computes $\xi_{KA}(w_{auA})^{\tilde{r}_{uqB}}$ then obtains $\xi_{KA}(w_{auA}\tilde{r}_{uqB})$.
7 B computes $\xi_{KA}(\tilde{r}_{uqA})^{w_{auB}}$ then obtains $\xi_{KA}(w_{auB}\tilde{r}_{uqA})$.
8 B computes $w_{auB}\tilde{r}_{uqB}$ and encrypts it with KA.
9 B repeats steps 6–8 replacing $\tilde{r}_{uqP}$ with proper c_{uqP}.
10 B adds up and finds $\xi_{KA}(w_{auA}\tilde{r}_{uqB} + w_{auB}\tilde{r}_{uqA} + w_{auB}\tilde{r}_{uqB})$ and $\xi_{KA}((w_{auA})_B + w_{auB})$ sends to A.
11 A decrypts them and adds $w_{auA}\tilde{r}_{uqA}$ to the former and $(w_{auA})_A$ to the latter.
12 A divides numerator by the denominator, adds a's mean, finds prediction p_{aq}.

4.1.4. Removing Overlaps

As seen from **Fig. 2**, in order to remove overlaps, there are two processes: eliminating initially filled votes (*smoothing*) and privately determining and removing overlaps (*privately match & remove)*. In the first step, each party deletes v_ds after preprocessing. Note that such v_ds are avoided to cause additional overlaps. In the second step, the problem is how to privately determine which ratings are overlapped. Such a problem can be deliberated as two parties having two sets and want to find commonly existing items. In privacy-preserving data mining, such problems are paid so much attention and some privacy-preserving set intersection protocols are proposed for parties having confidential data. In this context, Freedman et al. [26] presented some efficient schemes and in order to find overlaps, we prefer to apply one of them, namely *private matching for semi-honest parties(PM-Semi-Honest)*. PM-Semi-Honest scheme is a two-party protocol between *chooser* and *sender* both having different size of sets having numbers from the same domain. At the end of the protocol, chooser learns which of inputs are shared by both of them.

We propose privately matching and removing overlaps protocol (*Privately Match & Remove*) in order to tackle with overlaps. Initially, each party P finds indices of rated cells and computes cutting index point (λ_{ci}) where $\lambda_{ci} = (nm)/2$. Finally, each party P ends up with knowledge of approximately half of the total overlaps and deletes ratings held by P having indices corresponding such overlaps. After removing overlaps, parties move on to the next process *PrivateSims*. This solution is named as *ultimate scheme* (*US*). If the parties do not need or prefer to remove overlaps, *plain scheme* (*PS*), which does not involve overlap removing process, can be applied.

Privately Match & Remove: Privately matching and removing overlaps protocol

1 Each party P finds indices of rated cells and computes λ_{ci}
2 For rating index from the first to λ_{ci}
  2.1 Set A as chooser and B *as* sender
  2.2 Apply *PM-Semi-Honest*
  2.3 A learns about half of the overlaps and removes corresponding rating values

3 For rating index from λ_{ci} to the end
3.1 Switch parties' roles in steps 2.1–2.3, *B* removes remaining of the overlaps

4.2. Privacy-Preserving Item-based CF on Overlapped Ratings

Similar to subsection 4.1, we also propose two different schemes, as with the case of plain and ultimate ones in terms of privacy-preserving items-based CF. Such solutions are schematized as in **Fig. 3**. There are some common blocks with our user-based solution which are "Preprocessing" and "Privately Match & Remove". Such common blocks are the same as those presented in subsections 4.1.1 and 4.1.4. However, the remaining ones are are going to be mentioned in the following texts. In contrast to our user-based scheme, the ultimate scheme does not include preprocessing step in the item-based CF since preprocessing makes no sense for non-overlapping case of Slope-one algorithm.

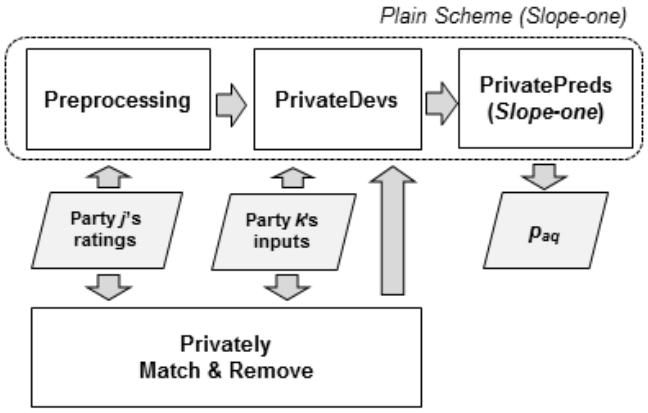

Fig. 3. Privacy-preserving item-based CF on overlapped ratings

4.2.1. Deviation Computation

Deviation computation given in Eq. 3 can be considered as in Eq. 5 and, similarly, it can be rewritten as

$$dev_{jk} = \frac{\sum_i (r_{ij} - r_{ik})}{card_{jk}} = \frac{\sum\sum_{i \in Z_{PQ}} (X_{Pi} - Y_{Qi})}{\sum_{\forall (P,Q)} |Z_{PQ}|} = \frac{\sum_{\forall (P,Q)} (R_{PQ_X} - R_{PQ_Y})}{\sum_{\forall (P,Q)} |Z_{PQ}|} = \frac{\sum_{\forall (P,Q)} (R_{PQ})}{\sum_{\forall (P,Q)} |Z_{PQ}|} \quad (7)$$

where X_P and Y_Q are column vectors consisting of r_{ij} and r_{ik} values held by party *P* and *Q*, respectively; Z_{PQ} stands for commonly rated users through vectors X_P and Y_Q. Hence, there are 4 different sub-components as a combination of $P = A$, $Q = A$, $P = A$, $Q = B$, etc. If $P = Q$, then numerator and dominator parts can be locally computed by each party. However, similar to our user-based scheme, the computation for cross sub-components is still challenging. Such challenge can be solved via private deviation computation protocol (PrivateDevs) as given below. In the PrivateDevs protocol, each part ends up with half of all $dev_{j,k}$ and $card_{j,k}$ values.

PrivateDevs: Private deviation computation protocol

1 Each party P assigns zero to unrated cells in X_P and Y_P.
2 For half of deviation values
3 For each item pairs (j, k)
3.1 Each party P computes $\sum(X_P - Y_P)$ and $|Z_{PP}|$
3.2 Each party P generates binary clone rating column vector (c_{jP}) and (c_{kP}).
3.3 Party A encrypts all X_A,-Y_A, c_{jA} and c_{kA} with its public key KA.
3.4 A sends $\xi_{KA}(X_A)$, $\xi_{KA}(Y_A)$, $\xi_{KA}(c_{jA})$, and $\xi_{KA}(c_{kA})$ to B.
3.5 B computes $\xi_{KA}(R_{AB_X}) = \prod_i \xi_{KA}(X_{Ai})^{c_{ikB}}$, $\xi_{KA}(R_{AB_Y}) = \prod_i \xi_{KA}(c_{ijA})^{Y_{Bi}}$, $\xi_{KA}(R_{BA_X}) = \prod_i \xi_{KA}(c_{ikA})^{X_{Bi}}$, and $\xi_{KA}(R_{BA_Y}) = \prod_i \xi_{KA}(Y_{Ai})^{c_{ijB}}$.
3.6 B computes $\xi_{KA}(|Z_{AB}|) = \xi_{KA}(c_{jA})^{c_{kB}}$ and $\xi_{KA}(|Z_{BA}|) = \xi_{KA}(c_{jB})^{c_{kA}}$
3.7 B obtains $\xi_{KA}(R_{BB}) = \xi_{KA}(\sum(X_B - Y_B))$ and $\xi_{KA}(|Z_{BB}|)$ by encrypting with KA.
3.7 B computes $\xi_{KA}(R_{AB})\xi_{KA}(R_{BA})\xi_{KA}(R_{BB})$, and $\xi_{KA}(|Z_{AB}|)\xi_{KA}(|Z_{BA}|)\xi_{KA}(|Z_{BB}|)$ sends these encrypted sub-aggregates to A.
3.8 A decrypts such encrypted sub-aggregates and adds $\sum(X_A - Y_A)$ and $|Z_{AA}|$ values to the corresponding sub-aggregates and obtains dev_{jk} and $card_{jk}$.
4 For the remaining deviation values
4.1 By switching their roles, B obtains such deviations and corresponding cardinalities.

4.2.2. Prediction Computation

Prediction computation is triggered with the prediction query "p_{aq}" of active user from MP whose rating profile is distributed among the parties. Deviations and cardinalities are also distributed among the parties. We need to privately compute Eq. 4 from the distributed elements. If we rearrange Eq. 4, then we obtain the following:

$$p_{aq} = \frac{\sum_j (dev_{qj} + r_{aj}) \times card_{qj}}{\sum_j card_{qj}} = \frac{\sum_j (num(dev_{qj}) + r_{aj} \times card_{qj})}{\sum_j card_{qj}} \tag{8}$$

where $num(x)$ is the numerator of x. To solve Eq. 8, parties use the protocol PrivatePreds (Slope-one) which privately computes prediction for the two parties. In this protocol, parties share encrypted version of held deviation values for item j, then the other party computes partial values of numerator and denominator of p_{aq} using homomorphic encryption properties. At the end of PrivatePreds (Slope-one), MP ends up with the final value of p_{aq} and inputs it a.

PrivatePreds (*Slope one*): Privately prediction computation protocol for Slope-one predictor

1 A informs B about p_{aq}
2 Each party computes partial $num(p_{aq})$ and $den(p_{aq})$ for $dev_{jk,}$ and r_{aj} is held by the

party.

3 Each party encrypts all held $num(dev_{jk})$ and $card_{jk}$ values related to item q and send it to the other party with its own public key.

4 A computes $\xi_{KB}(card_{qj_B})^{r_{aj_A}}$ $\xi_{KB}(num(dev_{qj_B}))$ and $\xi_{KB}(\sum_j card_{qj_B})$ for the r_{aj_A} values and sends these values to B.

5 B decrypts these partial values.

6 B computes $\xi_{KA}(card_{qj_A})^{r_{aj_B}}$ $\xi_{KA}(num(dev_{qj_A}))$ and $\xi_{KA}(\sum_j card_{qj_A})$ for r_{aj_B} and adds other available partial $num(p_{aq})$ and $den(p_{aq})$ values to this ciphertext and sends it to A.

7 A decrypts them and adds available corresponding partial data and obtains $num(p_{aq})$ and $den(p_{aq})$.

8 A divides $num(p_{aq})$ and $den(p_{aq})$. to find p_{aq} and returns it to a.

5. ANALYSIS AND SIMULATION RESULTS

5.1. Analysis of the Schemes

We proposed the two-fold PPCF solutions for two different CF approaches and then proposed schemes to make two parties collaborate on partitioned data with rating overlaps. First of all, we assert that our schemes met the privacy requirements mentioned in Subsection 3.4. Via randomized filling with default votes and homomorphic encryption, confidentiality of rated items and rating values are ensured. In the preprocessing step, we exploit v_ds to avoid share of actual sum, count, and sum of squares. Such v_ds improve privacy-preservation especially when there are a few number ratings for in a row (user). For instance to compute user mean values, each party P share disguised numbers such as $count+0.01\beta_P\delta_P$ rather than sharing actual $count$ values. From the side of other party Q, before trying to infer which items are rated, he should guess $count$ first. The probability of correctly guessing β_P is $1/\theta$ if β_P is considered integer. If β_P is considered rational number, this probability reduces with increasing precision of selection interval of $[0, \theta]$. However, at the same time, Q still have no certain information about density (δ_P) of P. One way to estimate $count$ values approximately, Q can analyze shared count values for the same users over number of trials where β_P is expected to be $\theta/2$. To avoid such kind of inferences, parties should scramble labels of users in a particular frequency of sharing. Note also that inference of individual rating values using disguised sum values is much more difficult than correctly guessing of which items are rated.

Default votes enhance privacy-preservation along the remaining procedures of plain schemes of both user- and item-based schemes as well. How about the proper values of default votes? v_ds can be row mean or column mean of held data. In particular, for our user-based CF scheme, column mean can be considered as more privacy enhancing solution since sum and count values of each row are shared among parties. However, for our item-based method, there is no exchange for local row or column-related values then row and column mean values can be considered as v_d. In addition to randomization provided by v_ds, we exploit cryptographic mechanisms as well in order to accomplish privacy-preservation. Paillier [28] proved that his homomorphic cryptosystem achieves semantic security for any

probabilistic polynomial time adversary. Privacy-preservation of our protocols *PrivateSims*, *PrivateDevs*, and *PrivatePreds* (for both cases) is directly based on such evidence. The privacy of *Privately Match & Remove* is fulfilled by Freedman et al's *PM-Semi-Honest* [29]. Their private matching protocol can be implemented based on Paillier's scheme or its subsequent versions hence privacy-preservation is based on the same proof. Also, self-blinding property of Paillier's homomorphic cryptosystem makes much more sense for a typical user-item data. There are numerous unrated cells and there are many cells expected to have the same value from a particular integer range, and such property effectively camouflages unrated and same-rated cells.

Since privacy and efficiency are two clashing goals, privacy-preservation mechanisms require additional communicational, computational and storage requirements. Using *PrivateSims*, for each similarity values, parties need to exchange $O(n)$ vectors with each other in two different communications. They can exchange such values of all similarities over just two communications: one from P to Q and one from vice versa. Similarly, for the case of PrivatePreds, O(m) vectors are exchanged between two parties and they can also be performed over two communications. We propose a distributed model in which similarity and deviation values are distributed between two parties. To compute each p_{aq}, parties need each other and one communication is needed from each party to other. To avoid prediction computation on distributed model, such similarity and/or deviation values can be entirely on each party depending on application.

Computational overheads are dominated by homomorphic operations. For *PrivateSims*, to compute each similarity value, there are totally $3m$ encryptions, $2m+1$ homomorphic multiplications and 1 decryption performed collaboratively by two party. For *PrivatePreds* based on user similarities, to compute each prediction value, MP need to perform $5n/2$ encryptions and 1 decryption while the other party perform $2(n+1)$ homomorphic multiplications. For *PrivateDevs*, to compute each deviation value there are totally $6n$ encryptions, $6n$ homomorphic multiplications and $4n$ homomorphic additions and 2 decryptions performed in collaboration of parties. To compute prediction based on item deviations by *PrivatePreds*, assuming that each party holds $m/2$ of deviations related to item q, each party is expected to perform m encryptions, m/2 homomorphic multiplications, m homomorphic additions and 2 decryptions. Considering large dataset where n and m are greater values, cryptographic operations may be bulky in computation, however recent research on implementation of efficient homomorphic encryption [30] shows that homomorphic encryption takes 24 ms, decryption takes at least 15 ms, addition is instantenous as taking as 1 ms whereas multiplication takes 41 ms on ordinary computer with 2.1 GHz Intel Core 2 duo processor with 1 GB of memory. With utilization of more powerful hardware infrastructures and parallel computation techniques, more satisfactory performance can be obtained. Also, to increase efficiency, some improvements such as pre-computation of normalization, similarity values and predictions before a's request may be possible. However, the parties must be ready for additional storage overheads in this case. For example, there will be requirement of $n^2/2$ of floating point number space.

5.2. Simulation Results

In our experiments, we use MLP datasets having ratings from 943 users for 1682 movies. It is collected by GroupLens research community and publicly available at their web site *www.grouplens.org*. There are in all 100,000 integer ratings from the domain of [1, 5]. In our experiments, we divide such available ratings into train and test subsets having 90% and 10% of available ratings randomly assigned to corresponding subsets, respectively. Ratings

in the train subsets are utilized to achieve CF algorithm and generate prediction while actual rating values in test subset are compared with predicted values to observe prediction quality in terms of accuracy. To evaluate accuracy, we use mean absolute error (MAE), which is popularly exploited in CF researches [3, 11]. MAE equals average of absolute differences between predicted values and actual test ratings. To reach dependable results, we perform 100 trials for each experiment and in each trial, train and test ratings are randomly determined. Each displayed MAE value is the average of MAEs obtained from all trials for each experiment.

First of all, we want to observe how ratio of overlaps changes with varying density of rating data and the level of filling. We perform trials by increasing δ from 10 to 100 and θ from 0 to 100 and demonstrate the percentages of overlaps in **Table 1**. Such percentage values reflect number of overlaps over the cardinality of union of ratings between both parties. When the data type is whole, the all available 100,000 ratings are taken into account and then the ratings are randomly selected. Else, such ratings are determined from train data consisting of 90,000 ratings. Note that when θ is 0 there is no filling, and when θ is 100 there may be default votes as much as actual ratings. According to **Table 1**, with increasing density overlapping ratio increases for all of the rows. However, such ratio is inversely proportional to θ since rating values can only be from fixed 90,000 cells while v_ds can be assigned to remaining cells, i.e., 1,496,126 cells.

Table 1. Ratio of Overlaps (%) vs. Density and Filling Level

Data Type	Filling (θ)	Density (δ)					
		10	20	40	60	80	100
Whole	0	5.28	11.11	24.99	42.86	66.67	100.00
Train	0	4.70	9.93	21.95	36.97	56.26	81.82
Train	10	4.47	9.40	20.67	34.72	52.24	75.29
Train	20	4.29	8.93	19.71	32.29	48.32	71.34
Train	40	3.94	8.11	18.08	29.69	43.22	61.87
Train	60	3.64	7.57	16.65	26.83	40.37	55.70
Train	80	3.49	7.13	15.50	24.88	36.22	49.89
Train	100	3.18	6.84	14.65	22.86	34.64	48.59

In the second experiment, we examine how accuracy changes with different levels of filling. For this reason, we vary θ from 10 to 100 and compute MAE values for PS and US for such θ values. Regarding the analysis in subsection 5.1, we select column mean as v_d for user-based CF scheme and row and column mean as v_d for item-based CF scheme. We set δ as 60 then each party holds 60% of ratings randomly selected from train subset and 36.97% of them are expected to be overlapped according to **Table 1**. For user-based CF algorithm, MAEs of PS and US with respect to varying θ are given in **Fig.4**. As seen from **Fig.4**, two schemes show different accuracy characteristics against increasing θ. While accuracy of PS worsen with the large level of filling, that of US gets better insignificantly, and US has the lowest MAEs for all θ values. The figure shows that θ does not affect accuracy of US as much as PS since US eliminates v_ds by the smoothing process. For each scheme, the best MAEs are 0.7513 and 0.7442 achieved at PS (θ = 20) and US (θ = 60), respectively.

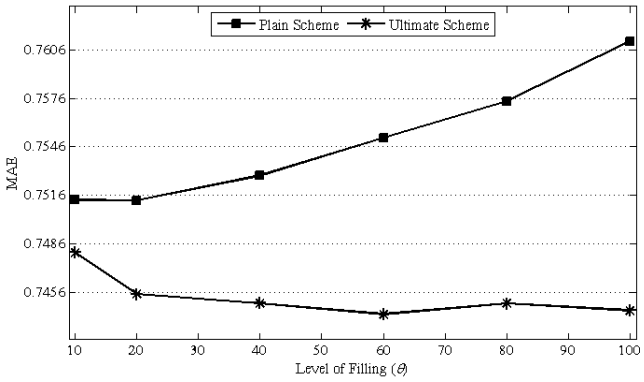

Fig. 4. Accuracy with respect to varying level of filling (User-based CF)

Similar to our user-based scheme, we also conduct some trials to evaluate change of accuracy with respect to varying level of filling for item-based CF. Since US does not include the preprocessing step, there no need to compare it with PS in terms of level of filling. Regarding the analysis in subsection 5.1, we use row and column mean as v_d for item-based CF scheme and display corresponding accuracy outcomes in **Fig.4**. According to **Fig.4**, row mean usage is slightly better than column mean for Slope-one CF, and both types provide worse accuracy with increasing amount of filling. Except for θ = 20, where the best MAE value is observed.

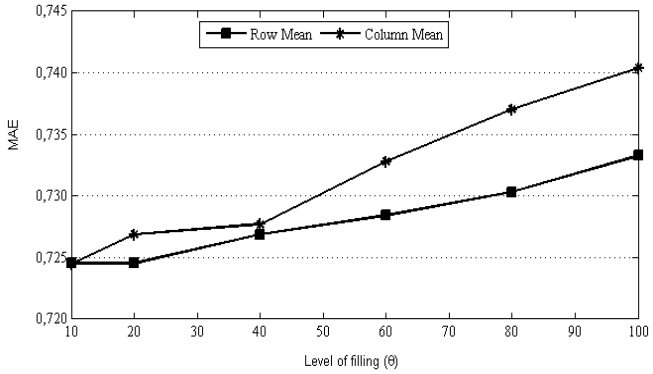

Fig. 5. Accuracy with respect to varying level of filling (Item-based CF)

In the context of our study, any party can produce prediction using three different methods: singly without collaboration and our schemes PS and US. In this set of experiments, we considered these three methods and compute MAEs for varying densities from 10 to 80. From **Fig. 4**, we set θ to optimum values 20 and 60 for PS and US, for user-based schemes, respectively. In **Table 2**, we display the outcomes related to user-based schemes and corresponding gains obtained by PPCF schemes with respect to single evaluation of CF in percentages where $Gain(X) = 100 \times (MAE_{Single} - MAE_X)/MAE_{Single}$ and MAE_X stands for the obtained MAE value from method X. According to **Table 2**, observed gains due to PPCF schemes get higher with lower densities. Hence, proposed user-based schemes work well for the parties having fewer amounts of ratings. This complies with our motivation which promote the prediction quality of the parties that suffer from data scarcity. We also check statistical significance of the results. For example, t-values of the results from PS and US are 47.60 and 31.14 , respectively, for δ = 20. For both t-values, the two-tailed P value is less

than 0.0001, and by conventional criteria the differences between single party and each of the user-based PPCF schemes are considered to be extremely statistically significant. The other *t*-values provide the same confidence level for promised accuracies by our schemes, except PS (δ = 60) and US (δ = 80). For PS (δ = 60), *t*-value is less than 1 and it can be said that it is not statistically significant. For US (δ = 80), *t*-value equals 2.96 and this means that the two-tailed P value is 0.0035 and by the way the difference caused by US can be said to be statistically very significant according to conventional criteria.

To evaluate overall performance of our item-based solutions, we conducted some experiments, and display obtained MAEs in **Table 3**. For PS, we fill data using optimum settings of θ = 20 and v_d; row mean is set according to **Fig.5**. Comparing to **Table 2**, gain values for item-based schemes are much greater than user-based schemes. Hence, our item-based schemes promise substantial contribution to accuracy especially for parties having sparse data. PS gives better accuracy than US especially for δ values of 20 and 40, and we can say that default votes and rating overlaps can be expected to contribute to the accuracy of CF. We can list two-tailed t-values for PS as {87.25, 69.47, 32.56, 12.03, 4.71} and US as {87.21, 52.87, 17.25, 1.76, 0.13 } for δ values of 10, 20, 40, 60, and 80, respectively. Such *t*-values show that the results are more statistically significant especially for lower values of δ. Another point is that the statistical signifance parameters of PS is greater than that of US despite of randomization-based mechanism in PS.

Table 2. Overall performance with varying density (User-based)

Method	**δ = 10**	**20**	**40**	**60**	**80**
Single Party	0.9265	0.8381	0.7729	0.7517	0.7416
Plain S.	0.8627	0.7936	0.7624	0.7513	0.7457
Ultimate S.	0.8798	0.8003	0.7562	0.7443	0.7391
Gain (PS)	6.88	5.31	1.35	0.06	-0.54
Gain (US)	5.04	4.51	2.16	0.99	0.34

Table 3. Overall performance with varying density (Item-based)

Method	**δ = 10**	**20**	**40**	**60**	**80**
Single Party	0.9936	0.8233	0.7613	0.7413	0.7378
Plain S.	0.7957	0.7416	0.7288	0.7292	0.7321
Ultimate S.	0.7955	0.7633	0.7455	0.7400	0.7394
Gain (PS)	19.91	9.93	4.27	1.63	0.78
Gain (US)	19.93	7.29	2.07	1.73	-0.02

6. CONCLUSIONS AND FUTURE WORK

In this work, we conceptually introduced the problem of rating overlaps between two parties in the context of some privacy-preserving collaborative filtering. We investigated the problem in terms of two different collaborative filtering approaches and proposed novel schemes. Such schemes come up with two alternative schemes such as the plain scheme and

ultimate scheme. While the plain scheme gives the de facto solution involving some privacy-preserving collaborative filtering process blocks without considering rating overlaps, ultimate scheme consists of such blocks and an overlap removing process. The empirical results show that our schemes contribute to the prediction quality of the parties while ensuring their privacy. Plain schemes for user-based or item-based collaborative filtering is very effective for lower data density.

We introduced overlapped ratings in partitioned data and our study is based on a conventional user-based collaborative filtering and Slope-one which is an effective item-based collaborative filtering method. Depending on the application, parties can utilize any four of our proposals to collaborate on overlapped data. Due to the opportunity of operation on overlapped ratings, our schemes promise a more practical setup over existing some privacy-preserving collaborative filtering solutions.

As a future study, more complicated scenarios can be considered, as in this study since we simplified the problem by equalizing the overlapping entries; however, in practice, much more complex overlapping cases could be faced. It is worth to examine such cases in the privacy-preserving manner. In this study, we considered just two parties, but there are some e-commerce sites that collaborate with multiple parties. This is also another interesting topic to focus in further research.

References

[1] A. Koschmider, T. Hornung, and A. Oberweis, "Recommendation-based editor for business process modeling," *Data & Knowledge Engineering*, vol. 70, no.6, pp. 483-503, June, 2011. Article (CrossRef Link).

[2] J.H. Park, "A recommender system for device sharing based on context-aware and personalization," *KSII Transaction on Internet and Information Systems*, vol. 4, no. 2, pp. 174-190, April 2010. Article (CrossRef Link).

[3] J.L. Herlocker, J.A. Konstan, A. Borcher, and J.T. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proc. of 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 230-237, August 15-19, 1999. Article (CrossRef Link).

[4] J.A. Calandrino, A. Kilzer, A. Narayanan, E.W. Felten, and V. Shmatikov, "You might also like: Privacy risks of collaborative filtering," in *Proc. of 2011 IEEE Symposium on Security and Privacy*, pp. 231-246, May 22-25, 2011. Article (CrossRef Link).

[5] J. Xiong, Z. Yao, J. Ma, X. Liu, Q. Li, J. Ma, "PRIAM: Privacy preserving identity and access management scheme in cloud," *KSII Transaction on Internet and Information Systems*, vol. 8, no. 1, pp. 282-304, January 2014. Article (CrossRef Link).

[6] I. Gunes, A. Bilge, and H. Polat, "Shilling attacks against memory-based privacy-preserving Recommendation Algorithms," *KSII Transaction on Internet and Information Systems*, vol. 7, no. 5, pp. 1272-1290, May 2013. Article (CrossRef Link).

[7] N. Lathia, S. Hailes, and L. Capra, "Private distributed collaborative filtering using estimated concordance measures," in *Proc. of 1st ACM conference on Recommender Systems*, pp. 1-8, October 19-20, 2007. Article (CrossRef Link).

[8] I. Yakut and H. Polat, "Arbitrarily distributed data-based recommendations with privacy," *Data & Knowledge Engineering*, February 2012, vol. 72, pp. 239-256, February 2012. Article (CrossRef Link).

[9] A. Basu, J. Vaidya, and H. Kikuchi, "Efficient privacy-preserving collaborative filtering based on the weighted Slope One predictor," *Journal of Internet Services and Information Security*, vol. 1, no. 4, pp. 26-46, November 2011.

[10] Su, X. and T.M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, p. 2-21, January 2009. Article (CrossRef Link).

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proc. of 10th international conference on World Wide Web*, pp. 285-295, May 1-5, 2001. Article (CrossRef Link).
[12] D. Lemire and A. Maclachlan, "Slope one predictor for online rating-based collaborative filtering," in *Proc of 2005 SIAM International Conference on Data Mining*, pp. 471-475, April 21-23, 2005. Article (CrossRef Link).
[13] M. Papagelis and D. Plexousakis, "Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents," *Engineering Applications of Artifical Intelligence*, vol. 18, no. 7, pp. 781-789, October 2005. Article (CrossRef Link).
[14] I. Yakut and H. Polat, "Estimating NBC-based recommendations on arbitrarily partitioned data with privacy," *Knowledge-based Systems*, vol. 36, pp. 353-362, December 2012. Article (CrossRef Link).
[15] J. Kim, C. Park, J. Hwang, and H. J. Kim, "Privacy Level Indicating Data Leakage Prevention System," *KSII Transaction on Internet and Information Systems*, vol. 7, no. 3, pp. 558-575, March 2013. Article (CrossRef Link).
[16] H. Polat and W. Du, "Privacy-preserving top-N recommendation on horizontally partitioned Data," in *Proc. of 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 725-731, September 19-22, 2005. Article (CrossRef Link).
[17] P.K. Prasad and C.P. Rangan, "Privacy preserving BIRCH algorithm for clustering over arbitrarily partitioned databases," *Lecture Notes on Computer Science*, vol. 4632, pp. 146-157, August 2007.Article (CrossRef Link).
[18] H. Polat and W. Du, "Privacy-preserving collaborative filtering on vertically partitioned data," *Lecture Notes on Computer Science*, vol. 3721, pp. 651-658, October 2005. Article (CrossRef Link).
[19] C. Kaleli and H. Polat, "Providing naïve Bayesian classifier-based private recommendations on partitioned data," *Lecture Notes in Computer Science*, vol. 4702, pp. 515-522, September 2007. Article (CrossRef Link).
[20] I. Yakut and H. Polat, "Privacy-preserving SVD-based collaborative filtering on partitioned data," *International Journal of Information Technology and Decision Making*, vol. 9, no. 3, pp. 473-502, May 2010. Article (CrossRef Link).
[21] C.L. Hsieh, J. Zhan, D. Zeng, and F. Wang, "Preserving privacy in joining recommender systems," in *Proc. of International Conference on Information Security and Assurance*, pp. 561-566, April 24-26, 2008. Article (CrossRef Link)
[22] J. Zhan, I.C. Wang, C.L. Hsieh, T.S. Hsu, C.J. Liau, and D.W. Wang, "Towards efficient privacy-preserving collaborative recommender systems," in *Proc. of IEEE International Conference on Granular Computing*, . 2008.Article (CrossRef Link) .
[23] C. Kaleli and H. Polat, "SOM-based recommendations with privacy on multi-party vertically distributed data," *Journal of Operational Research*, vol. 63, pp. 826-838. June 2012. Article (CrossRef Link).
[24] C. Kaleli and H. Polat, "Privacy-preserving trust-based recommendations on vertically distributed data," in *Proc. of Fifth IEEE International Conference on Semantic Computing*,pp. 376-379, September 18-21, 2011. Article (CrossRef Link).
[25] Y. Zhao, X. Feng, J. Li, and B. Liu, "Shared collaborative filtering," in *Proc. of the fifth ACM conference on Recommender systems*, pp. 29-36, October 23-27, 2011. Article (CrossRef Link).
[26] G. Jagannathan and R.N. Wright, "Privacy-preserving distributed k-means clustering over arbitrarily partitioned data," in *Proc. of 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 593-599, August 21-24, 2005. Article (CrossRef Link).
[27] A. Bilge, C. Kaleli, I. Yakut, I. Gunes, and H. Polat, "A survey of privacy-preserving collaborative filtering schemes," *International Journal of Software Engineering and Knowledge Engineering*, vol. 23, no. 8, pp. 1085-1108, October 2013. Article (CrossRef Link).
[28] P. Paillier, "Public key cryptosystems based on composite degree residuosity classes," *Lecture Notes in Computer Science*, vol. 1592, pp. 223-238, May 1999. Article (CrossRef Link).

[29] M. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," *Lecture Notes in Computer Science*, vol. 3027, pp. 1-19, May 2004. Article (CrossRef Link).
[30] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?," in *Proc. of the 3rd ACM Workshop on Cloud Computing Security Workshop*, pp. 113-124, October 21, 2011. Article (CrossRef Link).

Burak Memis is a Research Assistant in Department of Computer Engineering, Dumlupinar University, Kutahya, Turkey. He received the B.S. degree in Computer Engineering from Anadolu University, Eskisehir, Turkey in 2011 and is still master's candidate in Computer Engineering in Anadolu University, Eskisehir, Turkey. His areas of research include computer programming and data mining.

Ibrahim Yakut is an Assistant Professor in Dept. of Computer Engineering in Anadolu University, Eskisehir, Turkey while currently being visiting research fellow at MSIS Dept. of Rutgers University, Newark, USA. He received his M.S. and PhD. degrees in Computer Engineering from Anadolu University in 2008 and 2012, respectively. His research interests are recommender systems and privacy-preserving data mining.