Distributed Collaborative Filtering Protocol Based on Quasi-homomorphic Similarity

Hiroaki Kikuchi
Yoshiki Aoki
Graduate School of Engineering,
Tokai University,
4-1-1 Kitakaname, Hiratsuka, Kanagawa 259-1292, Japan.
kikn@tokai.ac.jp, ringo@cs.dm.u-tokai.ac.jp

Masayuki Terada
Kazuhiko Ishii
Kimihiko Sekino
NTT DOCOMO,INC.
3-6 Hikarinooka, Yokosuka-shi, Kanagawa,239-8536, Japan.
{*teradama,ishiikaz,sekino*}*@nttdocomo.co.jp*

***Abstract*—We study the problem of predicting the rating for an unseen item based on distributed dataset by two honest-but-curious parties without revealing each private dataset. Our proposed idea uses a new similarity measure such that similarity aggregated with two local similarities is approximately equal to the global similarity. We show the accuracy reduction and the performance gain given by our proposed scheme based on an experimental implementation, and claim that our scheme allows parties to estimate prediction in a practical model with negligible accuracy reduction.**

***Keywords*-Collaborative Filtering; Privacy-Preserving Data Mining; Cryptographical Protocol;**

I. Introduction

1) Background: Privacy preserving data mining aims to allow computation of useful aggregate statistics over the entire dataset without compromising the privacy of individual data. Parties wish to collaborate in obtaining aggregate results, such as the recommendation systems [2], the Naive Bayes classifier [3], the association rule mining [8], but may not fully trust each other. Parties may be competitors in the same field or be not allowed to exchange their customer's dataset by their privacy policies.

Vertically partitioned data is an important data distribution model often found in real life. For example, Table I illustrates two datasets partitioned vertically and owned by party A with attribute A_1 and A_2; and party B with attribute A_3 and a target class C, indicating whether or not to play tennis on a particular day. Parties A and B collect the different features, e.g. temperature, humidity or windy, but on the same day. Collaboratively performing Naive Bayes classification allows them to predict accurately if tennis is to be played or not; i.e. predict C given A_1, A_2 and A_3, however they can not share other's dataset partitions.

Vaidya and Clifton presented the secure protocol for Naive Bayes classifier to the vertically partitioned dataset without revealing individual data [3]. Their protocol combines the homomorphic public-key encryption algorithm to compute scalar product of two vectors and the secure function evaluation [10] for comparison of class c_y in C in terms of conditional attributes, i.e. $Pr(C = c_y | A_1 = a_1, A_3 = a_3)$.

Table I
SYNCHRONOUSLY (VERTICALLY) PARTITIONED DATASET

	party A		party B	
day	A_1	A_2	A_3	C
1	sunny	hot	high	no
2	sunny	hot	low	yes
3	rainy	hot	high	yes
4	rainy	cool	low	yes

2) Our Goal: In this work, we focus on the privacy-preserving protocol for Collaborative Filtering (CF) [16], a method to estimate the recommendations of unseen items based on the preference of communities of those who have evaluated the target items and have the similar preferences with the user who wish to get the recommendation. Canny [14] uses an additive homomorphic cryptosystem to perform Singular Value Decomposition (SVD) of the matrix of ratings on items by users. In [21], Ahmad and Khokhar studied the modified version of Canny's protocol using the modified ElGamal cryptosystem instead of the Paillier cryptosystem [15]. Katzenbeisser and Petkovic proposed an application to consumer healthcare services using the cryptographic private profile matching techniques based on homomorphic encryption algorithms [20].

All existing attempts, however, have been made only on horizontally partitioned datasets. Vertically partitioned datasets involve the transfer of confidential data between multiple enterprises, and thus is not trivial. Therefore, we introduce a public-key algorithm with additive homomorphic property, which allows us to performe collaborative filtering. However, the naive implementation has the following drawbacks;

1) the large number of ciphertexts are generated for every subsets of the user set of size n, and
2) the scalability with increasing number of users.

In order to address the above difficulties, we propose the followings;

1) CF with Quasi-homomorphic similarity

allows to compose local similarities which approximates the global similarity with small errors,
2) Pre-computation
helps to save encryption time of rating values in multiple recommendation queries,
3) k-Nearest Neighbour
works for reducing the number of ciphertexts to be sent to parties distributed over the network.

We evaluate the proposed schemes using a sample implementation with a public dataset, MovieLense[1], and show that the schemes are efficient and accurate to predict the ratings.

II. Preliminaries

A. Model

Let $U = \{u_1, u_2, \ldots, u_n\}$ be a set of users, where n is the number of users. Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items, where m be the number of items. Let $r_{y,j}$ be a rating given by user u_y for item i_j, for $y = 1, \ldots, n$, and $j = 1, \ldots, m$. Domain of rating value is 0, 1,.., 5, where 5 is the highest and 0 indicates missing value. Discrete value is used to evaluate items. Users do not evaluate all items. We denote a missing rating by $r_{y,j} = \phi$.

We assume that the matrix of ratings contains many missing elements, and thus it is a sparse matrix.

The goal of CF is to predict a missing rating of an item based on the other users' preferences to the given item. Our model supposes that users are willing to get recommendations for items that they are not seen before, but at the same time they are concerns privacy of rating made by themselves.

B. User-based Collaborative Filtering Algorithm

Collaborative Filtering (CF) is an algorithm to estimate missing ratings based on the preferences database. The prediction for user u for item o is given by a weighted average of users whose ratings are similar to the target user: $\hat{r}_{u,o} = \frac{\sum_{v \in U-\{u\}} s_{u,v} r_{v,o}}{\sum_{v \in U-\{u\}} s_{u,v}}$,where $s_{u,v}$ is the similarity between users u and v. The weight $s(u_y, u_j)$ is the similarity measure between users u_y and u_j, such as the Pearson correlation coefficient, or the Euclidean distance. In this work, we use a similarity measure defined as

$$s_{u,v} = \frac{1}{1 + \sum_{i \in I_u \cap I_v} (r_{u,i} - r_{v,i})^2},$$

where I_u is the set of items which rated by u-th user.

C. Homomorphic Encryption

To preserve the privacy of users, we use a public-key cryptosystem E which satisfies an additively homomorphic property, i.e. taking message M_1, M_2, $E[M_1]E[M_2] = E[M_1 + M_2], E[M_1]^{M_2} = E[M_1 M_2]$. For instance, the Paillier cryptosystem[15] and the modified ElGamal cryptosystem are widely used. Both allow us to get key generation and decryption processes distributed among semi-trusted authorities sharing private key. And we can decrypt by decryption function D. For instance, $D[E[M_1]] = M_1$.

The Paillier is more efficient than the ElGamal in the sense of decryption overhead, while the latter requires a sort of brute force technique (in the limited domain) for decrypting candidates of messages. We implement the Paillier cryptosystem for performance evaluation since the computational cost for a single encryption is significant in our proposed protocol.

III. Proposed Scheme

A. Overview

Our protocols use *Quasi-homomorphic similarity* which allows us to compose local similarities which approximates the global similarity with a small error. Pre-computation technique helps to save on time for encryption of rating values. Finally, the k-Nearest Neighbour works for reducing the number of ciphertexts to be sent to parties distributed over a network.

B. Naive CF

Let us recall the prediction of rating for item o and user u, $\hat{r}_{u,o}$, given by

$$\begin{aligned} \hat{r}^{AB}_{u,o} &= \frac{\sum_{v \in U-\{u\}} r_{v,o}/(1 + r^A_v + r^B_v)}{\sum_{v \in U-\{u\}} 1/(1 + r^A_v + r^B_v)} \\ &= \frac{\sum_{v \in U-\{u\}} r_{v,o} \prod_{\ell \neq v} (1 + r^A_\ell + r^B_\ell)}{\sum_{v \in U-\{u\}} \prod_{\ell \neq v} (1 + r^A_\ell + r^B_\ell)}, \end{aligned} \tag{1}$$

where

$$r^A_v = \sum_{i \in I_A} (r_{v,i} - r_{u,i})^2. \tag{2}$$

Note that performing of prediction consists of (1) local computation, e.g. r^A_ℓ, r^B_ℓ, and (2) joint computation with A and B, e.g. $r^A_\ell r^B_\ell$. The former can be locally performed, while the latter needs interaction between the two parties, which can be performed employing secure scalar product protocol. We show a cryptographical protocol for computing the prediction of $r^{AB}_{u,o}$ without revealing ratings as Algorithm 1.

C. Quasi-Homomorphic Similarity

The notion of similarity s satisfies $s_{u,u} = 1$ (*idempotent*), $s_{u,v} = s_{v,u}$ (*commutative*), and $s_{u,o} + s_{o,v} \geq s_{u,v}$ (*transitive*). For distributed computation, we prefer the definition of similarity that satisfies *homomorphic* property. Namely, mapping s is homomorphic if and only if there exists a function f such that

$$s(\boldsymbol{a}_u \cup \boldsymbol{b}_u, \boldsymbol{a}_v \cup \boldsymbol{b}_v) = f(s(\boldsymbol{a}_u, \boldsymbol{a}_v), s(\boldsymbol{b}_u, \boldsymbol{b}_v))$$

for any $\boldsymbol{a}_u, \boldsymbol{a}_v$ and $\boldsymbol{b}_u, \boldsymbol{b}_v$. $\boldsymbol{a}_u, \boldsymbol{a}_v$ are vector of u-th user's items. Those items are element of I_A. Similarly, $\boldsymbol{b}_u, \boldsymbol{b}_v$ are

Algorithm 1 Naive CF

Input: A's ratings $r_{v,i}$ $i \in I_A$, B's ratings $r_{v,i}$ $i \in I_B$
Output: predicted ratings $\hat{r}_{u,o}$

1) Party A computes ratings r_v^A locally for all i in I_A according to Eq. (2).
2) Similarly, party B computes ratings r_v^B locally for all i in I_B.
3) B sends $E(r_{v,o}\prod_{\ell\neq v} r_\ell^B)$ and $E(r_{v,o})$ to A.
4) A computes $y = \prod_{v\in U} E(\prod_{\ell\neq v} r_\ell^B r_{u,o})^{r_v^A}$ and $z = \prod_{v\in U} E(\prod_{\ell\neq v} r_\ell^B r_{u,o})$, after that A sends y and z back to B.
5) B has the prediction for user u and item o as $r_{u,o}^{AB} = D(y)/D(z)$.

elementof I_B. For instance, a cardinality $s(u,v) = |I_A \cap I_B|$ has a function $f(x,y) = x + y$, and squared Euclidean distance $s(\boldsymbol{a},\boldsymbol{b}) = ||\boldsymbol{a}-\boldsymbol{b}||^2$ satisfies homomorphic, though distance does express the degree of anti-similarity. Well-known similarity measures such as Pearson correlation coefficient and cosine similarity are not homomorphic.

Fully homomorphic similarity is hard to define. Instead, we define *quasi-homomorphic* similarity as one that the global similarity can be composed of the local similarities. Formally, a similarity s is quasi-homomorphic if and only if there exists small constant ϵ such that

$$|s(\boldsymbol{a}_u \cup \boldsymbol{b}_u, \boldsymbol{a}_v \cup \boldsymbol{b}_v) - f(s(\boldsymbol{a}_u, \boldsymbol{a}_v), s(\boldsymbol{b}_u, \boldsymbol{b}_v))| \leq \epsilon$$

for any $\boldsymbol{a}_u, \boldsymbol{a}_v \in I_A$ and $\boldsymbol{b}_u, \boldsymbol{b}_v \in I_B$. In a later section, we use an inverse squared Euclidean distance as the quasi-homomorphic similarity in order to perform prediction of items from vertically partitioned datasets.

A *normalisation* of similarity $s_{u,v}$ is defined as $\tilde{s}_{u,v} = \frac{s_{u,v}}{\sum_{\ell\in U-\{u\}} s_{u,\ell}}$. In a vertically partitioned dataset, a user-item ratings matrix is divided into two matrices owned by parties A and B. A *local similarity* is evaluated from only the item set owned by party A, and is written as $s_{u,v}^A$. Similarly, $s_{u,v}^B$ is the local similarity computed for only the item set owned by party B. On the contrary, we often say *global* similarity computed over the entire dataset.

D. Basic Scheme

The drawback of the Naive scheme is the complexity. Expanding Equation (2) generates all possible subsets of the set of users, which results in $O(n^2)$ ciphertexts to be sent, where n is the number of users. Hence, we propose a light-weight prediction scheme replacing the global similarity with the composition of two local similarities as follows: $\hat{r}_{u,o}^{A*B} = r_{u,o}^A \cdot w_A + r_{u,o}^B \cdot w_B$, where w is a weight defined by $w_A = \frac{m_A}{m_A+m_B}$, and m_A is $|I_A|$, m_B is $|I_B|$.

Correspondingly, the overall prediction $\hat{r}_{u,o}$ for user u and item o is given by $\hat{r}_{u,o} = \frac{\sum_{v\in U-\{u\}} s'_{u,v} r_{v,o}}{\sum_{v\in U-\{u\}} s'_{u,v}}$, where s' is normalised local similarity, i.e., $s'_{u,v} = \frac{s_{u,v}}{\sum_{l\in U-\{u\}} s_{u,l}}$. The Algorithm 2 gives the Basic Scheme.

Algorithm 2 Basic Scheme

Input: A's ratings $r_{v,i}$ $i \in I_A$, B's ratings $r_{v,i}$ $i \in I_B$
Output: prediction $\hat{r}_{u,o}^{A*B}$ for user u and item o.

1) Party A computes local normalized similarity $\tilde{s}_{u,v}^A$ of user u for every user $v \in U - \{u\}$.
2) Party B computes local normalized similarity $\tilde{s}_{u,v}^B$ of user u for every user $v \in U - \{u\}$, encrypts with B's public key and then send to A ciphertexts $E(r_{1,o}), \ldots, E(r_{u-1,o}), E(r_{u+1,o}), \ldots, E(r_{n,o})$.
3) A computes $y = E(r_{1,o})^{\tilde{s}_{u,1}^A} \cdots E(r_{n,o})^{\tilde{s}_{u,n}^A}$, and sends back to B.
4) B decrypts y and performs the prediction of item o for user v as

$$\hat{r}_{u,o}^{A*B} = \frac{m_B}{m} D(y) + \frac{m_A}{m} \sum_{v\in U-\{u\}} r_{v,o}\tilde{s}_{v,o}^B, \quad (3)$$

where $\tilde{s}_{v,o}^B$ is the local normalised similrity evaluated by B.

(例 III.1) We show local and global similarities in Table II.

Table II
LOCAL AND GLOBAL SIMILARITIES

	s_A	s_B	s_{AB}	s_{A*B}
u_1	0.839	0.294	0.750	0.512
u_2	0.076	0.117	0.100	0.101
u_3	0.083	0.588	0.150	0.386
u_4	-	-	-	-

Note that Equation (3) can be rewritten as

$$\hat{r}_{u,o}^{A*B} = \hat{r}_{u,o}^A \frac{m_A}{m} + \hat{r}_{u,o}^B \frac{m_B}{m},$$

which implies the weighted sum of two local predictions, $\hat{r}_{u,o}^A$ and $\hat{r}_{u,o}^B$, according to the ratio of m_A and m_B.

E. Pre-computation Scheme

The basic scheme requires to perform $n-1$ encryptions, which increases with an increase in the number of users. In order to address the heavy computational overhead for encryption, we present an efficient scheme with pre-computation of ratings.

The idea is based on the fact that the domain of encryption is limited within the set of ratings. The range of values in a typical dataset is $\mathcal{D} = \{1,2,3,4,5\}$. Therefore, pre-computing all possible encryptions of rating values saves the processing time for encryption. Namely, the n computations is reduced up to 5 computations, $\mathcal{D} = \{E(1), E(2), E(3), E(4), E(5)\}$. These are not secure because the same rating values have the same encryptions. In

Algorithm 3 Pre-comutation Scheme

Input: A's rating values $r_{v,i}$ $i \in I_A$, B's rating values $r_{v,i}$ $i \in I_B$.

Output: prediction $\hat{r}_{u,o}$ for user u and item o.

1) (pre-computation step) Party B encrypts all elements in the domain of rating, $\mathcal{D}$, and encrypts 0 for p times. Let Z be the set of "zero"ciphertexts.
2) (prediction step) B generates ciphertext for plain rating value x as $E(x) = c_x \cdot d$, where c_x is the x-th ciphertext in $\mathcal{D}$ and d is uniformly chosen from Z. The rest of prediction are as the same as the basic scheme.

Algorithm 4 k-Nearest Neighbor Scheme

Input: A's rating values $r_{v,i}$ $i \in I_A$, B's rating values $r_{v,i}$ $i \in I_B$.

Output: k-th prediction $\hat{r}_{u,o}^{A*B_k}$ for user u and item o.

1) Same as Step 1 in Basic Scheme.
2) B sorts the set of users $U - \{u\}$ in order of normalized local similarity $\tilde{s}_{u,v}^{B}$ and choose the k highest users, letting $U(u,k)^B$ be the subset of U. Then, B performs normalization in $U(u,k)^B$ so that the local similarities $\tilde{s}_{u,v}^{B_k}$ in $U(u,k)^B$ sum up 1.0 and for each $v \in U(u,k)^B$, and sends encryptions $(v, E(r_{v,o}))$ to A.
3) For each of $U(u,k)^B$, A computes $y = \prod_{v \in U(u,k)^B} E(r_{v,j})^{\tilde{s}_{u,j}^{A_k}}$ and sends it back to A.
4) Finally, B decrypts y and predicts the target rating value in conjunctions the normalized local similarities $\tilde{s}_{v,o}^{B}$ in Equation (3).

order to make ciphertexts for the same ratings indistinguishable from each other, we multiplies a "zero" ciphertext of 0 to the ciphertexts, i.e. $E(0) \cdot E(x)$ gives new ciphertext $E(x)'$ whose plaintext remains unchanged. The zero ciphertext is chosen from the set of ciphertext $Z = \{E(0)_1, \ldots, E(0)_p\}$.

The Algorithm 3 shows the overall steps for the pre-computation technique.

F. k-Nearest Neighbour Scheme

The prediction of rating value is based on the similarities between users. Hence, the prediction using the k nearest users can improve the performance of prediction without losing accuracy. The *k-Nearest Neighbour* selection is one of the well-known techniques in collaborative filtering and is defined in Algorithm 4.

IV. Evaluation

A. Performance in Trial Implementation

We implemented our proposed schemes on Java SDK version 1.6 with `BigInteger` class. Our trial implementation performs interaction between a server and a client in the same local area network. Table III[1] shows the performances for fundamental cryptographical primitives that were run on MacOS X (Core 2 Duo 2.26 GHz, 4GB RAM). We use the performance constants, i.e. ℓ_c, t_e and t_d, to estimate the total performance with large dataset.

Table IV shows the communication costs for the proposed schemes. In the table, ℓ_c is the size of ciphertext, typically $2,048$ bit. Note that k-NN reduces the size up to k and can be used with the pre-computation scheme.

Table V shows the number of operations; encryption, multiplication and exponentiation for each of the proposed schemes. Since the overhead of encryptions dominates the overall performance, the pre-computation technique would help in significant reduction of processing time.

Table III
Performance of our trial implementation

attribute	value
public-key algorithm	2048 bit Paillier
size of ciphertext	$\ell_c = 256$ [byte]
encryption time	$t_e = 160$ [ms]
decryption time	$t_d = 248$ [ms]
exponentiation time	$t_m = 0.093$ [ms]
multiplication time	$t_m = 0.102$ [ms]

B. Processing and communication cost

The experimental dataset to evaluate the practical performance of the proposed scheme is "MovieLens"[1] with $n = 943$ users, $m = 1,682$ items, and $100,000$ evaluted raiting values. We use random vertical partitioning such that A and B have equal size of portions of the dataset.

We show the processing time with respect to number of users n in Figure 1. The Basic scheme takes 152 seconds at $n = 900$, while the Pre-computation scheme runs in 4.45 seconds, which is 34-fold times improvement against the Basic scheme.

Table IV
Communication Costs

scheme	B to A	A to B
2. Basic	$\ell_c(n-1)$	ℓ_c
3. Precomputation	$\ell_c(n-1)$	ℓ_c
4. k-NN	$\ell_c(k-1)$	ℓ_c
Precomputation&k-NN	$\ell_c(k-1)$	ℓ_c

Table V
Computation costs for each step

scheme	$E(M)$	$E(M_1) \cdot E(M_2)$	$E(M_1)^{M_2}$
2. Basic	$n-1$	$n-1$	$n-1$
3. Precomputation	0	$2(n-1)$	$n-1$
4. k-NN	$k-1$	$k-1$	$k-1$
Precomputation & k-NN	0	$2(k-1)$	$k-1$

[1]The exponents are limited within relatively small numbers in CF and hence the processing time is extremely smaller than that of an ordinary modular exponentiations with exponent chosen from full domain.

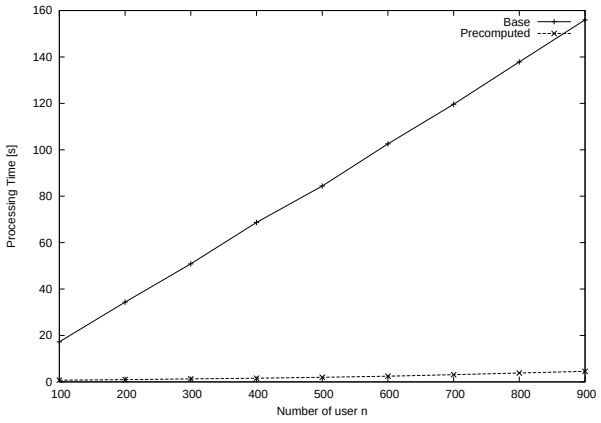

Figure 1. Processing time with respects to number of users n in Basic and Precomputation schemes

The communication is linear to the number of users in the matrix.

C. Accuracy

We evaluate the accuracy of proposed schemes for predicting 100 values randomly chosen out of 100,000 rating values in the dataset. The schemes are (1) *Global NN CF* – k-NN CF applied to the joint datasets, (3) *Aggregated NN CF* – k-NN CF applied to the composite dataset from two partial datasets. We shows the experimental results of the Mean Absolute Error (MAE) with respect to k for four schemes in Figure 2. Note that $k = n$ corresponds the results of the Basic scheme.

The MAE decreases as k, size of rating values used for prediction, increases. The minimum MAE is 0.9595 at $n = 942$, where aggregated scheme has MAE of 0.9588 with distance of 0.007. The mean errors are reduced with even smaller set of the size k, hence the k-NN performs well for accuracy.

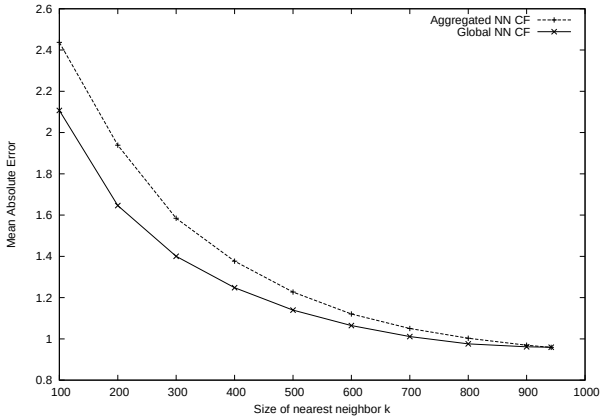

Figure 2. Accuracy with respect to k neighbours,for global and aggregated datasets

The source of MAE come from the inconsistency of the composite (approximate) similarities and the global (true) ones. Let us recall the composite similarities defined as $\hat{s} = s_A \cdot w_A + s_B \cdot w_B$, where s_A and s_B are local similarities evaluated for partitioned datasets by parties A and B, respectively. The weights, w_A and w_B, adjust the skew of partition, particularly $w_A = w_B = 1/2$ in our experiment.

In order to compare the correlation between local and global similarities, we show ths scatter plot between these similarities in Figure 3.

From the scatter plot, we observe a weak positive correlation between the global and the composite similarities. The Pearson correlation coefficient is 0.673, which varies for the target users. The possible reason of inconsistency includes the skew of the target user, the distribution of rating values, and the digit of finite precision.

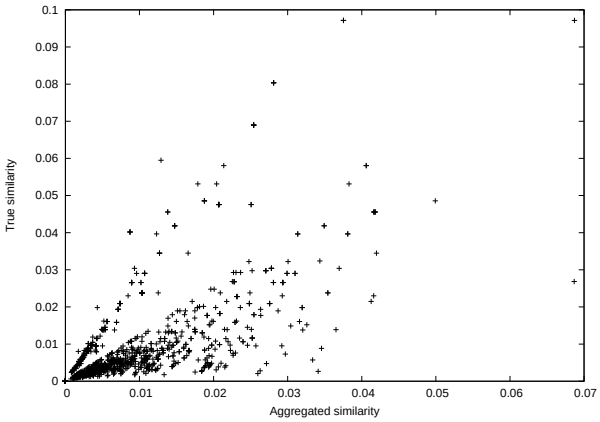

Figure 3. Scatter plot between the global and the composite (aggregated) similarities

D. Security

The confidentiality of rating values is preserved based on the security of public-key algorithm used to encrypt values. Hence, party A has no chance to obtain the B's private rating in all of proposed schemes since what (s)he learns from the execution of schemes is the n ciphertexts sent from party B.

However, B has some advantage to learn something from B's rating since (s)he eventually learns the result of decryption of y, which yields the sum of the product of rating and A's local similarities, i.e., $r_{1,o}\hat{s}^A_{u,1} + \cdots + r_{n,o}\hat{s}^A_{u,n}$, where $r_{v,o}$ can be replaced with arbitrary values. For instance, manipulated vector $(1, 0, \ldots, 0)$ allows A to learn $\hat{s}^A_{u,1}$.

In this paper, we assume "semi-honest model", in which parties follow the protocols as specified but are curious about what the counterpart has. Hence, the above malicious behavior is not expected to be exhibited in our scenario. In order to prevent malicious A from intentionally manipulating rating values, we need to add the zero-knowledge proof

protocol to ensure that vector contains enough entropy to make the identification to B's local similarity impossible. This is one of the future challenges of this work.

V. Conclusion

In this paper, we proposed the series of schemes for collaborative filtering from vertically partitioned datasets. Our proposed schemes are efficient against the naive but perfect privacy collaborative filtering protocols in terms of computation and communication costs, from $O(n^2)$ to $O(n)$, and $O(k)$ in k-Nearest Neighbor scheme. The schemes preserve the accuracy of prediction of ratings, which can be improved with the number of ciphertexts to send, k. Our experimnets show that the pre-computation technique saves the computation time by 34 times with the size of 900 users and the MAE is 0.9588 for the composite similarity.

Our future works include the study with variety of similarities from the view point of preserving privacy, improving efficiency and the comprehensive evaluation of collaborative filtering algorithms.

References

[1] Grouplens Data Sets, (`http://grouplens.org/`)

[2] Haifeng Yu, Chenwei Shi, Kaminsky, M., Gibbons, P.B., and Feng Xiao, "DSybil: Optimal Sybil-Resistance for Recommendation Systems", in IEEE Symp. on Security and Privacy, pp. 283-298, IEEE, 2009.

[3] Jaideep Vaidya and Chris Clifton, "Privacy preserving naive bayes classfier for vertically partitioned data", In 2004 SIAM International Conference on Data mining, pp. 522-526, 2004.

[4] Wenliang Du and Mikhail J. Atallah, "Privacypreserving statistical analysis", In Proceeding of the 17th Annual Computer Security Applications Conference, pp. 10-14 2001.

[5] Yehuda Lindell and Benny Pinkas, "Privacy preserving data mining", Journal of Cryptology, 15(3), pp. 177-206, 2002.

[6] Rakesh Agrawal, Alexandre Evfimievski, and Ramakrishnan Srikant, "Information shareing across private databases", in proc. of ACM SIGMOD International Conference on Management of Data, 2003.

[7] Michael J. Feedman, Kobbi Nissim, and Benny Pinkas, "Efficient private matching and set intersection", in Eurocypt 2004, IACR, 2004.

[8] Jaideep Vaidya and Chris Clifton, "Secure set intersection cardinality with application to association rule mining", Journal of Computer Security, Vol. 13, No. 4, pp. 593-622, 2005.

[9] G. Jagannathan and R. N. Wright, "Privacy-Preserving Distributed k-Means Clustering over Arbitrarily Partitioned Data", *ACM KDD05*, 2005.

[10] Andrew C. Yao, "How to generate and exchange secrets", In Proc. of the 27th IEEE Symposium on Foundations of COmputer Science, pp. 162-167, 1986.

[11] Dahlia Malkhi, Noam Nisan, Benny Pinkas, and Yaron Sella, "Fairplay – A Secure Two-Party Computation System", Usenix Security Symposium, 2004.

[12] Koji Chida, Dai Ikarashi, and Katsumi Takahashi, "Tag-Based Secure Set-Intersection Protocol and Its Application", in proc. of Computer Security Symposium (CSS 2009), IPSJ, 2009 (in Japanese).

[13] L. F. Cranor, "I Didn't Buy it for Myself, Privacy and E-Commerce Personalization", WPES 2003, Washington, DC, USA, pages 111-117, 2003.

[14] J. Canny: Collaborative Filtering with Privacy, *IEEE Conf. on Security and Privacy*, Oakland CA, May 2002.

[15] P. Paillier: "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes", Proc. *EUROCRYPT'99*, LNCS 1592, pp. 223-238, 1999.

[16] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering", In UAI, pp. 43-52, 2004.

[17] Resnick, P., Iacovou, N., Sushak, M., Bergstrom, P., and Riedl, J. , "GroupLens: An open architecture for collaborative filtering of netnews", Proceedings of the 1994 Computer Supported Collaborative Work Conference.

[18] Amazon.com, (http://www.amazon.co.jp/ .)

[19] G. Morohash, et.al, "Secure Multiparty Computation for Comparator Networks", IEICE Trans. Fundamentals, Vol. E91-A, No. 9,2008.

[20] Katzenbeisser, S. and Petkovic, "Privacy-Preserving Recommendation Systems for Consumer Healthcare Services", In Proceedings of the 2008 Third international Conference on Availability, Reliability and Security (ARES 2008), IEEE Computer Society, pp. 889-895, 2008.

[21] Ahmad, W. and Khokhar, "An Architecture for Privacy Preserving Collaborative Filtering on Web Portals", In Proceedings of the Third international Symposium on information Assurance and Security, IEEE Computer Society, pp. 273-278, 2007.

[22] J. S. Breese, D. Heckrman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," In UAI, pp.43-52, 2004.

[23] H. Kikuchi, H. Kizawa and M. Tada, "Privacy-Preserving Collaborative Filtering Schemes", WAIS 2009, ARES 2009 federated workshop, IEEE Press, 2009.

[24] Grouplens Data Sets, `http://grouplens.org/`

[25] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl "Item-Based Collaborative Filtering Recommendation Algorithms," ACM WWW10, Hong Kong, May 2001.