A Privacy-Preserving Collaborative Filtering Protocol Considering Updates

Yuji Mochizuki and Yoshifumi Manabe
Department of Informatics
Kogakuin University
1-24-2 Nishishinjuku Shinjuku-ku Tokyo Japan
Email: em14015@ns.kogakuin.ac.jp, manabe@cc.kogakuin.ac.jp

***Abstract*—This paper proposes a method to update the similarity of items in a privacy preserving collaborative filtering. The similarity of items is a value that shows how similar given two items are. Privacy preserving collaborative filtering is a technique that helps to infer an evaluation value of desired items given the other users' evaluation values with concealing personal information for each user's privacy by encrypting the evaluation values. In order to obtain the most appropriate evaluation value, it is necessary to update the similarity every time when an evaluation value is changed. Since each evaluation value is encrypted, it is a heavy burden for the users to update the similarity of items every time in response to a single change of evaluation values. Thus we need to know when we should recalculate the similarity of items while keeping each renewal of evaluation value secret. In this paper, we show that the estimation error of an evaluation value is small if the error of the average of evaluation values between the users is small. We show an algorithm that detects a change of the average of the evaluation value that is greater than the preset, using a constant number of plaintext equality tests for each renewal of evaluation values.**

I. Introduction

A collaborative filtering(CF) is a technique to recommend an item(book, movie, etc) to users. CFs mainly use the similarity of users or the similarity of items. Many works proposed CFs with concealing user's private information[1][2]. Tada et al. proposed a CF using the similarity of items for the following three reasons[3]. 1)Similarity of items does not have user's private information and the cost for the computing the similarity of items is low. 2)The similarity of items is a characteristic common to all users, and if the similarity of items is once exposed, it is possible to calculate all estimation values using the similarity of items. 3)In most cases CFs using the similarity of items are more accurate methods than the ones using the similarity of users. Tada et al. stated that it is possible to continue to use the same similarity of items that is once published. Their scheme has a problem that each user's evaluation values on items vary as time goes on and the similarity values also vary along with them. This paper proposes a method to update the similarity of items in a privacy preserving collaborative filtering using the similarity of items.

II. Definitions

$s(i_k, i_l)$ denotes the similarity of item k and item l that is calculated by the following equation. Here $r_{j,k}$ denotes the evaluation value of an item k by user j, and n denotes the number of all users. $r_{j,k} = 0$ if user j has not evaluated the item k yet.

$$s(i_k, i_l) = \frac{\sum_{j=1}^{n} r_{j,k} r_{j,l}}{\sqrt{r_{1,k}^2 + ... + r_{n,k}^2}\sqrt{r_{1,l}^2 + ... + r_{n,l}^2}}$$

We define an estimation of evaluation value $P_{j,k}$ of item k for user j by the following formula.

$$P_{j,k} = \overline{r}_k + \frac{\sum_{l \in I_j} s(i_k, i_l)(r_{j,l} - \overline{r}_l)}{\sum_{l \in I_j} s(i_k, i_l)} \quad (1)$$

Here $\overline{r}_k$ denotes the average of the evaluation values for item k. The set of all items is denoted by I. We denote $I_j = \{k \in I | r_{j,k} \neq 0\}$.

The encryption used in this paper is additive homomorphic and has a threshold decryption property. Examples of such cryptosystem are Paillier[4] and modified ElGamal. In order to decrypt a ciphertext, modefied ElGamal needs a brute force algorithm to search for its plaintext. Since the range of the plaintexts is large in our proposed method, Paillier cryptosystem fits our method, hence we adopt Paillier cryptosystem in this paper. In Paillier cryptosystem, the secret key can be distributed to multiple users. When more than the threshold users cooperate, they can decrypt ciphertexts. In this method, multiple representative users conduct to decrypt if necessary. $E(x)$ denotes an encryption of plaintext x. Anyone can calculate $E(m_1 + m_2)$ from $E(m_1)$ and $E(m_2)$ without the decryption key.

Plaintext Equality Test (PET) is a technique to detect whether two ciphetertexts are generated by the same plaintext[5]. $ptest(a, b)$ denotes a PET for ciphetertexts a and b. If a and b are ciphetertexts that are generated by the same plaintext, $ptest(a, b)$ returns 0.

III. Privacy Preserving Collaborative Filtering

We use Tada et al.'s collaborative filtering[3]. v denotes the set of evaluation values that the users can evaluate, for instance in 5 point rating system $v = \{1, 2, 3, 4, 5\}$. $max(v)$ denotes the maximum evaluation value and $min(v)$ denotes the minimum evaluation value. In this example $max(v) = 5$ and $min(v) = 1$.

Users can evaluate estimation value $P_{j,k}$ for item k and user j with the following procedure without revealing each user's evaluation values. m denotes the number of items.

1. Each user $j = 1, ..., n$ calculates the following values for items $k = 1, ..., m$.

$$A_{j,k} = E(r_{j,k}), B_{j,k} = E(e_{j,k}), C_{j,k} = E(r_{j,k}^2),$$

where $e_{j,k}$ is the flag that is defined as follows: $e_{j,k} = 0 \ if \ r_{j,k} = 0, \ e_{j,k} = 1 \ otherwise$. User j evaluates $D_{j,k,l} = E(r_{j,k}r_{j,l})$ for all $k \in I$ and $l(\neq k) \in I$. Each user publishes $A_{j,k}, B_{j,k}, C_{j,k}$, and $D_{j,k,l}$ in the encrypted manner to all users.

2. Representative users calculate the following values for item $k = 1, ..., m$ to get the average of evaluation values and the norm. $n_k = |I_k|$ and $\overline{r_k}$ denotes the average of user's evaluation values for item k.

$$E(n_k\overline{r}_k) = \prod_{j=1}^{n} A_{j,k} = E(\sum_{j=1}^{n} r_{j,k})$$

$$E(n_k) = \prod_{j=1}^{n} B_{j,k} = E(\sum_{j=1}^{n} e_{j,k})$$

$$E(n_k\overline{r_k^2}) = \prod_{j=1}^{n} C_{j,k} = E(\sum_{j=1}^{n} r_{j,k}^2)$$

Also, representative users evaluate the following for all k and l.

$$\prod_{j=1}^{n} D_{j,k,l} = E(\sum_{j=1}^{n} r_{j,k}r_{j,l})$$

3. Representative users decrypt ciphertexts $E(n_k\overline{r}_k), E(n_k), E(n_k\overline{r}_k^2)$, and $E(\sum_{j=1}^{n} r_{j,k}r_{j,l})$.

4. Representative users evaluate the averages of evaluation values, norms and the similarities of each item, and publish the results to all users.

$$\overline{r}_k = \frac{n_k\overline{r}_k}{n_k} \tag{2}$$

$$||r_k|| = \sqrt{r_{1,k}^2 + \cdots + r_{n,k}^2}$$

$$s(i_k, i_l) = \frac{\sum_{j=1}^{n} r_{j,k}r_{j,l}}{||r_k||||r_l||} \tag{3}$$

5. User u_j calculates the estimation values by Equation 1 using the values obtained by Equations 2 and 3.

IV. The Proposed Update Scheme

Since user's preference changes with time, the value $r_{j,k}$ also changes with time and it affects the estimation. It is necessary for users to re-evaluate the similarity of items to reflect the variations on estimation values at every renewal of user's evaluation value to obtain the most relevant estimation. However it is inefficient to update the entire values since the amount of the calculation is very large against the changes caused by a single renewal. In addition, there is a risk that the other users can easily guess which item's evaluation value has been updated or even the updated value by comparing the new and old values n_k or $\overline{r_k}$. Thus the period of recalculation of the similarity of values must be long enough to conceal the user privacy and short enough to obtain relevant estimations, however it is difficult to assess the period since the evaluation values are encrypted. A periodic updete is a simple method, however it is not effective since the evaluation values might not be changed so much by the updates. We propose a scheme to detect the appropriate update timing.

When a user renew the evaluation value of an item, he performs procedure 1, and after that representative users perform procedure 2. When the change of the average of the evaluation values exceeded the predefined threshold, the fact is detected by procedure 2, and every user publishes new values of $E(r_{j,k})$, $E(e_{j,k})$, $E(r_{j,k}^2)$, and $E(r_{j,k}r_{j,l})$. Note that for the values that have no changes, re-encryption is done by adding E(0) to the old values. Using these values, each user can calculate new estimation values.

A. Procedure 1

We set variable X_k that has an encrypted value for each item $(k = 1, ..., m)$. The initial value of X_k is $E(0)$. We define the parameter Y_k as follows.

$$Y_k = n_k\overline{r}_k + d_kn_k$$

d_k is set to acceptable fluctuation $|\overline{r_k} - \overline{r_k}'|$ on the average of the evaluation value of the item k. $\overline{r_k}'$ denotes the average of evaluation value of item k that reflects all renewals. d_k is set so that $d_k + \overline{r_k}$ is an integer. If a user renew an evaluation value of an item, $\widetilde{r_{j,k}}$ has the difference of the new and old evaluation values, i.e., the initial value of $\widetilde{r_{j,k}}$ is set to 0 at the time of the calculation of the estimation values.

$$\widetilde{r_{j,k}} \leftarrow r'_{j,k} - r_{j,k}$$

In addition $f_{j,k}$ is defined as follows. $f_{j,k} = 1 \ if \ r_{j,k} = 0, \quad f_{j,k} = 0 \quad otherwise$

procedure 1

$For \ \ k = 1 \ \ to \ \ m$
  $if \ \ \widetilde{r_{j,k}} \neq 0$
    $E(X_k) \leftarrow E(X_k) + n_kE(\widetilde{r_{j,k}}) - f_{j,k}E(Y_k)$
  $otherwise \quad E(X_k) \leftarrow E(X_k) + E(0)$
$r_{j,k} \leftarrow r'_{j,k}$

B. Procedure 2

procedure 2

$For \ \ S = 1 \ \ to \ \ max(v)$
  $ptest(E(X_k), E(N_kn_k + Sn_k))$

Representative users execute procedure 2 to detect whether the average of the evaluation values have been changed more than the threshold. In the procedure 2, N_k denotes the integer nearest to d_kn_k. They also need to perform procedure 2 for negative direction using a negative threshold. Assuming d'_k as the negative threshold value, it results $d'_k \neq -d_k$ since $d'_k + \overline{r_k}$ should be set as an integer.

If procedure 2 detects changes that exceed the threshold for some percentage or more items, it is necessary to re-calculate

the average of the evaluation values, $\overline{r_k}'$, and so on for the item.

V. PROOF OF DETECTION

We show that when the change of the average of the evaluation values exceeds the threshold, the fact can be detected with $2max(v)$ time PETs in procedure 2. In procedure 1, X_k is the following value from the calculation.

$$X_k = \sum_{x_{k,renew}} (n_k \widetilde{r_{j,k}} - f_{j,k} * Y_k)$$

$x_{k,renew}$: the set of users who renewed the evaluation value of item k.

We bound the maximum and minimum change of X_k by a single renewal. Note that $MAX(x)$ denotes the maximum value of x and $MIN(x)$ denotes the minimum value of x.

(1) When $f_{j,k} = 1$

$$MAX(\widetilde{r_{j,k}}) = max(v), MIN(\widetilde{r_{j,k}}) = min(v)$$

$n_k\overline{r_k}$ satisfies $n_k\overline{r_k} \leq n_k max(v)$. We assume that d_k is at most $\pm min(v)$. We denote ΔX_k as the difference of X_k between before and after a renewal of an evaluation value. Since $-min(v) \leq d_k \leq min(v)$, and thereby $0 \leq Y_k \leq (max(v)+1)n_k$ is satisfied. Thus it holds $-max(v)n_k \leq \Delta X_k \leq max(v)n_k$ and ΔX_k mod $n_k = 0$.

(2) When $f_{j,k} = 0$

$MAX(r'_{j,k}) = max(v)$, $MIN(r_{j,k}) = min(v)$ thus $MAX(\widetilde{r_{j,k}}) = max(v) - min(v)$.

$MIN(r'_{j,k}) = min(v), MAX(r_{j,k}) = max(v)$ thus $MIN(\widetilde{r_{j,k}}) = min(v) - max(v)$

These values satisfy $-max(v)n_k \leq \Delta X_k \leq max(v)n_k$ and ΔX_k mod $n_k = 0$.

In either case, ΔX_k at one renewal satisfies $-max(v)n_k \leq \Delta X_k \leq max(v)n_k$ and $\Delta X_k \ mod \ n_k = 0$. Thus, when X_k changes more than the threshold, the fact can be detected by the PETs, because the PETs are executed between $E(X_k)$ and $E(n_k(N_k + S))$.

VI. EVALUATION OF PROPOSED ALGORITHM

$P'_{j,k}$ denotes the estimation value that is calculated using all updated values. $s'(i_k, i_l)$ denotes the similarity between item k and item l assuming that all renewals have been performed. I'_j denotes the set of items that include renewals, thus $I'_j \supseteq I_j$.

$$P'_{j,k} = \overline{r'}_k + \frac{\sum_{l \in I'_j} s'(i_k, i_l)(r'_{j,l} - \overline{r'}_l)}{\sum_{l \in I'_j} s'(i_k, i_l)}$$

The difference of the estimation value before and after updates, $P_{j,k} - P'_{j,k}$ can be calculated as follows.

$$P_{j,k} - P'_{j,k} = \overline{r}_k - \overline{r'}_k + \frac{\sum_{l \in I_j} s(i_k, i_l)(r'_{j,l} - \overline{r}_l)}{\sum_{l \in I_j} s(i_k, i_l)}$$

$$- \frac{\sum_{l \in I'_j} s'(i_k, i_l)(r'_{j,l} - \overline{r'}_l)}{\sum_{l \in I'_j} s'(i_k, i_l)}$$

From the above equation, the term $\overline{r_k} - \overline{r_k}'$ influences most on the estimation value. The fact is verified by a simulation.

We show the errors of average of the evaluation values and the estimation value using "MovieLensDataSets" distributed by GroupLens[6]. It is a set of 100,000 values for 943 users of 1,682 items. We executed the following experiment 3,000 times. (1) 30,000 data is set as the users' initial evaluation values. (2)1,000 data are treated as the renewals of the evaluation values. (3)We verified the errors of estimation values and the change of the average of the evaluation values that is more than the threshold. The result is shown in TABLE 1 where S_1 : the number of trials, S_2 : the number of times when 0.07 or more variation was observed in both of the average of the evaluation value and the estimation of the evaluation value, S_3 : the number of times when 0.07 or more variation was observed only in the average of the evaluation values, and S_4 : the number of times when 0.07 or more variation was observed only in the estimation of the evaluation values. Precision is 0.78, Recall is 0.72, and F value is 0.75.

TABLE I. NUMBER OF TIME OF DETECTION

S_1	S_2	S_3	S_4
3,000 times	63 times	18 times	25 times

TABLE II. COMPUTATIONAL COMPLEXITY

	naive	our scheme
(every user) encryption	$T(\frac{m(m-1)}{2} + 3m)$	$T + \frac{m(m-1)}{2} + 3m$
(representative user) decryption	$T(\frac{m(m-1)}{2} + 3m)$	$5T + \frac{m(m-1)}{2} + 3m$
computational complexity	$O(m^2T)$	$O(T + m^2)$

We evaluate the computational complexity in TABLE 2 where T : the number of renewals since the last update. The computational complexity of the proposed method is better than the one of computational complexity of the naive method, in which the recalculation is executed at every renewal.

Our proposed method is not accurate enough to detect the exact error because of the difference between N_k and $d_k n_k$. We will try to compose an indicator that shows more accurate error of the estimation value or a new type of indicator that shows the timing when the average of the evaluation values and the similarity of items should be updated.

ACKNOWLEDGEMENT

This work was supported by JSPS KAKENHI Grant Number 26330019.

REFERENCES

[1] Canny J, "Collaborative Filtering with Privacy," Proceedings of the 2002 IEEE Symposium on Security and Privacy, pp.45-57 (2002).

[2] Ahmad W. and Ashgaq K., "An Architecture for Privacy Preserving Collaborative Filtering on Web Portals," Proc. 3rd International Symposium on Information Assurance and Security, pp.273-278 (2007).

[3] Tada M. and Kikuchi H., "Proposal on Privacy-Preserving Collaborative Filtering Protocol Based on Similarity between Items," Journal of Information Processing, vol. 51, no. 9, pp.1554-1562 (2010).

[4] Paillier P., "Public-Key Cryptosystems Based on Conposite Degree Residuosity Classes Advances," EUROCRYPT, LNCS, vol.1592, pp.223-238 (1999).

[5] Lipmaa, H., "Verifiable homomorphic oblivious transfer and private equality test," ASIACRYPT , LNCS, Vol.2894, pp.416-433 (2003).

[6] Grouplens Data Sets : http://www.grouplens.org/ (2014).