I. Terzi & H. Polat, Int. J. of Design & Nature and Ecodynamics. Vol. 11, No. 3 (2016) 284–294

PRIVACY-PRESERVING NORMALIZED RATINGS-BASED WEIGHTED SLOPE ONE PREDICTOR

I. TERZI & H. POLAT
Computer Engineering Department, Anadolu University, Eskisehir, Turkey.

ABSTRACT
Weighted Slope One predictor is proposed as a model-based collaborative filtering algorithm based on user ratings. The predictor is able to efficiently provide accurate predictions. The scheme utilizes user's true ratings. In this paper, we propose to utilize normalized user ratings like z-scores for the weighted Slope One predictor. Also, in order to protect privacy, we propose a privacy-preserving weighted Slope One predictor based on z-scores using randomization. Moreover, we utilize masked deviations to show how it affects accuracy of the proposed scheme. We perform various real data-based experiments to evaluate the overall performance of the proposed method. Empirical outcomes show that the algorithm is able to provide accurate predictions.
Keywords: accuracy, collaborative filtering, privacy, randomization, slope one, z-scores.

1 INTRODUCTION

In order to overcome *information overload* problem, collaborative filtering (CF) algorithms are widely used. CF is a filtering method utilizing other people's ratings about various products. The term "collaborative filtering" was first coined by the Tapestry project [1].CF assumes that if two customers have agreed in the past, they would tend to agree in the future. To conduct CF, n users' preferences about m products as ratings are collected. Then, $n \times m$ user-item matrix is created. This matrix is a very sparse matrix [2].

Generally speaking, CF algorithms have some problems like accuracy, privacy, efficiency, robustness, cold start, and so on [3]. The most important problems can be considered as accuracy, efficiency, and privacy. It is very important for CF schemes to provide accurate predictions efficiently while preserving privacy [4]. Due to increasing number of privacy risks posed by CF algorithms [4], users might hesitate to share their data with CF systems or refuse to provide data at all. Thus, preserving privacy is important in such systems. Without privacy protection, CF systems might not be able to collect enough data for filtering purposes because some users might refuse to give data. To offer accurate and dependable recommendations, CF systems should have sufficient data. Also, some users might tend to give false data rather than their true data due to privacy concerns. As a result, false data might lead to inaccurate predictions.

Privacy-preserving collaborative filtering (PPCF) schemes have been proposed to achieve privacy while providing accurate recommendations [2, 4]. Canny [5, 6] has initiated the collaborative filtering with privacy. Since then various studies on PPCF have been proposed in the literature [2, 4]. One of the most common privacy protection methods is known as randomization. Randomized perturbation techniques (RPTs) are widely used to mask confidential data in such a way so that CF systems are able to estimate accurate predictions while preserving privacy [7, 8]. RPTs generate random numbers using uniform or Gaussian distributions with zero mean. They are then added to confidential

WIT CONFERENCES
This paper is part of the Proceedings of the International Conference on Big Data (Big Data 2016)
www.witconferences.com

ISSN: 1755-7437 (paper format), ISSN: 1755-7445 (online), http://www.witpress.com/journals
DOI: 10.2495/DNE-V11-N3-284-294

data for data masking. Due to aggregate computations in CF, it is still possible to estimate accurate recommendations from perturbed data.

CF algorithms, in general, are grouped into memory-based, model-based, and hybrid algorithms [9]. Memory-based algorithms operate on entire user-item matrix. Thus, their online performance is poorer. Model-based algorithms create a model off-line and utilize such model online to provide predictions. Hence, their online performance is better. However, memory-based algorithms provide more accurate referrals than model-based methods. Hybrid algorithms try to combine the advantages of memory- and model-based schemes.

Lemire and Maclachlan [10] propose an efficient model-based CF algorithm known as Slope One predictor for estimating predictions on online ratings.Slope One predictors are popular candidates for CF systems. The proposed scheme is considered as a model-based CF scheme because deviations are computed off-line and are represented as a model. The scheme is able to efficiently provide accurate predictions. The authors present two variants of the Slope One predictors like weighted and bi-polar Slope One schemes. The scheme achieves comparable accuracy.

Although Slope One predictors are efficient and accurate schemes, they fail to protect individual user privacy. In order to preserve privacy in Slope One predictors, Basu *et al*. [11] propose a randomization-based privacy-preserving Slope One predictor. However, our proposed scheme here is different from their study as follows: First, they consider disguising user ratings only. However, in addition to ratings, rated/unrated items are also confidential and need to be masked. Second, we propose a Slope One predictor based on z-scores and perturb z-scores. Given z-scores, it is more difficult to derive ratings without knowing the related average ratings and the standard deviation of the ratings. Third, they also utilize encryption as a privacy measure, which is costly compared to randomization. Our scheme does not use any encryption for privacy protection.

In this paper, we modify the weighted Slope One predictor in such a way so that recommendations can be generated based on z-scores rather than ratings. We use RPTs to mask confidential data, ratings and rated/unrated items, of all users including the active user (the one who is looking for a prediction for a target item q).We also perturb the deviations as suggested by Basu *et al*. [11] for better performance. We perform real data-based experiments for evaluating the overall success of the proposed scheme.

2 RELATED WORK

PPCF has been receiving increasing attention since 2002. Canny proposed two CF schemes with privacy [5, 6]. His schemes are based on peer-to-peer architecture and utilize homomorphic encryption methods for privacy. The methods are based on aggregate data. Given aggregates and homomorphic encryption, deriving private data is difficult.Polat and Du [7, 8] propose to user RPTs for masking confidential data in CF systems. The authors claim that since CF is based on aggregate data, it is still possible to estimate accurate predictions from masked data. Polat and Du [12] and Yakut and Polat [13] discuss inconsistent data masking using RPTs. Since user privacy concerns might be different, each user might decide to perturb their data according to their concerns. The authors show that accurate recommendations can be estimated from inconsistently masked data.

Due to privacy measures, performance might become worse. To enhance online performance in PPCF schemes on RPTs, Bilge and Polat [14] propose dimensionality reduction-based PPCF scheme. The authors apply data reduction on perturbed data and provide predictions from reduced masked data. They disguise both ratings and rated/unrated items using randomization. Renckes *et al*. [15] suggest a hybrid PPCF method. The method is basically based on creating trees for each user in the user-item matrix off-line. The trees include neighbours and the related similarities estimated on perturbed data. The authors utilize RPTs as data-masking method. Chow *et al*. [16] present an

efficient PPCF scheme based on randomization. The method employs both hashing and clustering for improved online performance. The authors basically disguise the rated movies using randomization.Bilge and Polat [17] utilize bisecting *k*-means clustering to improve both accuracy and online performance while generating recommendations with privacy. Clustering improves performance while producing clones of users enhances accuracy.

Basu *et al*. [18, 19] study how to provide horizontally and vertically partitioned data-based predictions on Slope One predictors with privacy. They use additive homomorphic encryption to achieve privacy. The authors consider corporate privacy rather than individual user privacy. Their scheme protects two CF systems against each other. Our scheme is different from their scheme. They focus on partitioned data-based Slope One predictor while we scrutinize central server-based Slope One predictor with privacy. Basu *et al*. [11] propose a privacy-preserving Slope One predictor. The authors utilize both randomization and encryption for privacy. Their scheme is the closest study to our work here. However, our work is different from their study. We first propose z-score-based Slope One predictor without privacy concerns. For this purpose, we modify Slope One predictor proposed for rating-based CF [10]. We then mask both z-scores and rated/unrated items using RPTs. Basu *et al*. [11] mask ratings only. In addition to protecting ratings, revealing rated/unrated items might cause serious privacy risks. Hence, our scheme protects both confidential data. Moreover, since we disguise z-scores rather than ratings, even if z-scores are derived, malicious entity still needs average ratings and standard deviations to derive the ratings. Thus, our scheme can be considered more secure. They utilize encryption besides randomization. We use RPTs only as privacy measures. We finally scrutinize how masking deviation only affects accuracy.Gambs and Lolive [20]propose a semi-trusted third party-based Slope One predictor with privacy. Their scheme is based a semi-trusted party and randomized response techniques are utilized for privacy protection.

3 PRELIMINARIES

In this section, we explain the weighted Slope One predictor for rating-based CF [10]. A prediction for an active user a on a target item $q(p_{aq})$ can be estimated as follows:

$$p_{aq} = \frac{\sum_{j \in S} \left(dev_{qj} + v_{aj} \right) \times C_{qj}}{\sum_{j \in S} C_{qj}} \tag{1}$$

in which S is the set of rated items by a, v_{aj} is a's rating on item j, $C_{qj}$is the number of users who rated both items q and j, and dev_{qj} is the average deviation of item j with respect to item q.

The average deviation of item j with respect to item q can be estimated as follows:

$$dev_{qj} = \sum_{u \in S_{qj}} \frac{v_{uq} - v_{uj}}{C_{qj}} \tag{2}$$

in which S_{qj} is the set of users who rated both items q and j, v_{uq} and v_{uj} are the ratings of the user u on items q and j, respectively, and note that C_{qj} is the number of users who rated both items q and j.

In eqn (1), each value is weighted by the number of users who rated both items. Thus, the prediction is estimated as a weighted average. Since each item in user-item matrix can be selected as a target item,C_{qj} values for all $q = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, m$ can be computed off-line. Similarly, dev_{qj} values for all $q = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, m$ can be computed off-line. Therefore, dev_{qj} and C_{qj}valuescan be considered as prediction models and are generated off-line. During online phase, v_{aj}

values are used and predictions are estimated for active user a. Due to off-line model generation, online performance of the weighted Slope One predictor is better than traditional memory-based CF algorithms.

4 PRIVACY-PRESERVING SLOPE ONE ON NORMALIZED RATINGS

4.1 Problem definition

In this paper, we search solution for the following research questions: (*i*) The weighted Slope One predictor is based on user ratings. However, normalized ratings can be used for providing predictions. Thus, how can the weighted Slope One predictor be modified in such a way so that recommendations can be computed based on normalized ratings like z-scores? (*ii*) Users including active users are usually concerned about their privacy. The ratings they provided and their rated/unrated items are usually considered confidential data. Such private data should be protected. Hence, how can such confidential data be protected? And how can recommendations be estimated on perturbed private data? (*iii*) Deviations can be estimated by each user before they are sent to CF system. Since they are considered confidential, how can they be protected? And how does such approach affect accuracy?

4.2 Slope One predictor on z-scores

User preferences about various items are represented using numeric or binary ratings. The higher the numeric ratings, the more the user likes that item. The weighted Slope One predictor proposed by Lemire and Maclachlan [10] is based on user ratings. Normalization might improve accuracy [21]. Therefore, we modify the weighted Slope One predictor for providing predictions on normalized ratings. Each user normalizes their ratings by transforming them into z-scores. For a user u, given her rating v_{uj} for an item j, her average rating v_u, and standard deviation of her ratings σ_u, the related z-score (z_{uj}) can be computed as follows:

$$z_{uj} = \frac{v_{uj} - v_u}{\sigma_u} \tag{3}$$

After normalizing their ratings, users send them to the CF system. Then, user-item matrix is created to hold z-scores. Given z-scores, the CF system can estimate predictions using the following modified weighted Slope One predictor:

$$p_{aq} = v_a + \sigma_a \times \frac{\sum_{j \in S} \left(dev_{qjz} + z_{aj} \right) \times C_{qj}}{\sum_{j \in S} C_{qj}} \tag{4}$$

in which z_{aj} is a's z-score for item j, v_a is her average rating, σ_a is standard deviation of her ratings, and dev_{qjz} is average deviations based on z-scores. Note that since z-scores are used, the weighted average is de-normalized by multiplying it with σ_a and adding v_a to the result. Also, dev_{qjz} values based on z-scores can be computed as follows:

$$dev_{qjz} = \sum_{u \in S_{qj}} \frac{z_{uq} - z_{uj}}{C_{qj}} \tag{5}$$

4.3 Protecting confidential data using randomization

Confidential data consists of ratings and rated/unrated items. Note that ratings are normalized by transforming them into z-scores. To protect private data, users employ RPTs [14]. Random noise is added to user z-scores to mask real z-scores. Similarly, uniformly randomly selected some of the unrated item cells are filled with random numbers. Our data masking procedure can be summarized as follows:

1. Users normalize their ratings by transforming them into z-scores.
2. Each user u uniformly randomly select standard deviation of random numbers (σ_u) from the range $(0, \sigma_{max}]$, where σ_{max} is the upper bound of the standard deviations and is known as privacy parameter.
3. Users flip a coin to choose random number distribution like uniform or Gaussian.
4. Each user u uniformly randomly select an integer σ_u from the range $(0, \sigma_{max})$, where σ_{max} is the upper bound for the number of filled cells and is known as privacy parameter. Its value is dependent on the density of the users' ratings vectors.
5. Users compute σ_u percent of their unrated cells called f_u. Each user u then uniformly randomly selects f_u number of unrated cells to be filled.
6. Users generate random numbers using uniform or Gaussian distribution with 0 mean and σ_u. They are then added to the z-scores and chosen unrated cells to be filled.
7. Disguised data are finally sent to the CF system.

4.8 Estimating predictions on masked data

In this subsection, we show how to estimate recommendations from perturbed z-scores. Note that the CF system holds masked z-scores and active user *a*also masks her z-scores similarly. To get a prediction for q, a sends her disguised z-scores and her query to the server. Since v_a and σ_a are known by a, the server estimates the weighted average (called P_{aq}) using eqn (4) and sends it to a. She then de-normalize it and finds the related prediction. Thus, eqn (4) based on masked data can be written as follows:

$$\widehat{p_{aq}} = v_a + \sigma_a \times \frac{\sum_{j\in\hat{S}} \left(\widehat{dev_{qjz}} + \widehat{z_{aj}} \right) \times \widehat{C_{qj}}}{\sum_{j\in\hat{S}} \widehat{C_{qj}}} = v_a + \sigma_a \times P_{aq} \tag{6}$$

Note that due to filled unrated cells, $\hat{S}$ includes the rated and filled cells. Similarly, $\hat{C}$ includesthe number of users who rated both items and filled items by the same user. P_{aq} can be estimated as follows:

$$P_{aq} = \frac{\sum_{j\in\hat{S}} \left[\left(dev_{qjz} + R_{qjz} \right) + \left(z_{aj} + r_{aj} \right) \right] \times \widehat{C_{qj}}}{\sum_{j\in\hat{S}} \widehat{C_{qj}}} = \frac{\sum_{j\in\hat{S}} \left[\left(dev_{qjz} + z_{aj} \right) + \left(R_{qjz} + r_{aj} \right) \right] \times \widehat{C_{qj}}}{\sum_{j\in\hat{S}} \widehat{C_{qj}}} =$$

$$\frac{\sum_{j\in\hat{S}} \left(dev_{qjz} + z_{aj} \right) \times \widehat{C_{qj}}}{\sum_{j\in\hat{S}} \widehat{C_{qj}}} + \frac{\sum_{j\in\hat{S}} \left(R_{qjz} + r_{aj} \right) \times \widehat{C_{qj}}}{\sum_{j\in\hat{S}} \widehat{C_{qj}}} \approx \frac{\sum_{j\in\hat{S}} \left(dev_{qjz} + z_{aj} \right) \times \widehat{C_{qj}}}{\sum_{j\in\hat{S}} \widehat{C_{qj}}} \tag{7}$$

In eqn (7), R_{qjz} represents random noise due to average deviation estimations based on masked z-scores. The random numbers r_{aj}values are generated by*a* to perturb her z-scores. Since random numbers are generated using random number distributions with 0 mean and a standard deviation, the expected value of the second sum $\frac{\sum_{j\in\hat{S}}\left(R_{qjz}+r_{aj}\right)\times\widehat{C_{qj}}}{\sum_{j\in\hat{S}}\widehat{C_{qj}}}$ is zero. Thus, eqn (7) holds.

Like the predictions, average deviations should also be estimated based on disguised z-scores as follows:

$$\widehat{dev_{qjz}}=\sum_{u\in\widehat{S_{qj}}}\frac{\widehat{z_{uq}}-\widehat{z_{uj}}}{\widehat{C_{qj}}}=\sum_{u\in\widehat{S_{qj}}}\frac{\left(z_{uq}+r_{uq}\right)-\left(z_{uj}+r_{uj}\right)}{\widehat{C_{qj}}}=$$
$$\sum_{u\in\widehat{S_{qj}}}\frac{\left(z_{uq}-z_{uj}\right)+\left(r_{uq}-r_{uj}\right)}{\widehat{C_{qj}}}=\sum_{u\in\widehat{S_{qj}}}\frac{z_{uq}-z_{uj}}{\widehat{C_{qj}}}+\sum_{u\in\widehat{S_{qj}}}\frac{z_{uq}-z_{uj}}{\widehat{C_{qj}}}\approx\sum_{u\in\widehat{S_{qj}}}\frac{z_{uq}-z_{uj}}{\widehat{C_{qj}}} \quad (8)$$

Due to the same reasons, the expected value of the sum is zero. Hence, eqn (8) holds. In other words, average deviations can be estimated based on disguised z-scores. After estimating average deviations, P_{aq} can be estimated using eqn (7). The CF system or the server sends P_{aq} back to the user *a*. She finally de-normalizes it and finds the related prediction.

4.5 Analysis of the proposed scheme

Data masking using RPTs might cause additional costs. Such costs can be grouped as off-line and online costs. Compared to online costs, off-line costs are not that critical. Supplementary costs are classified as storage, communication, and computation costs. To store user preferences, an $n \times m$ user-item matrix is used. Even if disguised z-scores are sent, the system still needs the same matrix to store its users' data. Thus, our proposed scheme does not cause any extra storage costs.

The weighted Slope One predictor without privacy concerns requires two communications only. An active user *a* sends her ratings and a query to the server. After estimating a prediction, the server sends it back to *a*. In our scheme, number of communications is two only. Hence, our method does not cause any additional communication costs in terms of number of communications. Similarly, it does not cause any extra communication costs with respect to amount of transferred data because*a* sends her ratings vector and a query; and receives a prediction.

Due to filled unrated cells, computation costs are expected to increase. Note that number of filled cells depends on user ratings vector density. Such vectorsare very sparse. Even if the density increases by two times,on average, computation costs increase by two times only due to privacy measures.

The CF system tries to derive confidential data. It needs to figure out filled cells and determine true ratings. Privacy analysis of the utilized data-masking procedure can be done as described by Bilge and Polat [14]. The system receives filled normalized ratings vector from each user. The server first needs to guess the value of β given β_{max}. It then can guess the number of filled cells. Finally, it can guess the filled cells with a probability.

Since each user masks her z-scores, even if the server finds out z-scores, it becomes difficult to guess true ratings from z-scores because it does not know the related average rating and the standard

deviation of the ratings. Also, users randomly select random number distribution. Moreover, they uniformly randomly choose the standarddeviation for the random number distribution. Privacy levels introduced by randomization can be estimated as discussed in [14]. For example, when σ_{max} is 2, on average, σ_u is 1. Then, the privacy level provided by Gaussian distribution is about 3. With increasing σ_{max} values, privacy levels increase, too.

5 EXPERIMENTS

5.1 Data sets and evaluation criteria

To evaluate the overall success of our scheme, we performed experiments using two well-known real data sets. MovieLens Million (MLM) data set was collected by GroupLens (http://movielens.umn.edu/). It includes ratings of 6,040 users for 3,952 movies. The ratings discrete numeric ones ranging from 1 to 5. Netflix data set (http://www.netflixprize.com/) includes discrete numeric ratings ranging from 1 to 5, too. There are 480,189 users and 17,770 movies in the set.Due to larger number of users and movies, we selected a subset of the Netflix including 10,000 users and 4,000 movies using stratified sampling.

We used *mean absolute error* (MAE) as an accuracy metric because the rating ranges are the same for both data sets. MAE measures the average absolute deviation between the observed rating and the predicted rating. The smaller the MAE is, the better the scheme is. We also calculated the total amount of online time (*prediction time*-PT) in milliseconds (ms) to generate a prediction.

5.2 Methodology

We first uniformly randomly selected 1,000 users as test users for both data sets.We also uniformly randomly chose train users for both data sets. Note that the train and test sets are disjoint. For each test user, we estimated predictions for all of their rated items. We withheld a true rating and predicted its true value. We did this for all rated items. We then compared the withheld rating with the predicted one. Due to RPTs, we performed our experiments 100 times for privacy-preserving schemes and presented the overall averages.There are different control parameters that might affect the overall performance. The examples include σ_{max}, β_{max}, and number of train users (n). To show their effects, we varied the values of one parameter and fixed the values of the other parameters.

5.3 Experiments

We first conducted experiments to compare the proposed normalized ratings-based modified weighted Slope One predictor (MSO) with the oneproposed by Lemire and Maclachlan [10], known

Table 1: Comparison of the SO and MSO in terms of MAE.

n		**125**	**250**	**500**	**1,000**	**2,000**	**4,000**	**5,000**
MLM	SO	0.7650	0.7430	0.7392	0.7220	0.7181	0.7096	0.7103
	MSO	0.7582	0.7391	0.7219	0.7170	0.7082	0.7069	0.7065
Netflix	SO	0.8001	0.7791	0.7583	0.7553	0.7416	0.7350	0.7420
	MSO	0.7948	0.7738	0.7592	0.7502	0.7400	0.7406	0.7427

as Slope One (SO). We varied n from 125 to 5,000 for both data sets. After computing overall averages of MAEs, we displayed them in Table 1.

The results in Table 1 show that our scheme usually performs better than SO for both data sets. As seen from Table 1, normalization improves accuracy. With increasing n values, MAEs become better. However, when n becomes larger than 2,000, the results become stable and starts to become worse. Better results are observed for MLM. This is because MLM is denser than Netflix.

After comparing our normalization-based scheme with the Slope One, we scrutinized how varying σ_{max} values affect overall performance of MSO. Note that σ_{max} is one of the privacy control parameters. With increasing σ_{max} values, randomness augments. As a result, privacy improves. However, we hypothesized that increasing randomness makes accuracy worse. To verify this hypothesis, we performed a set of experiments while varying σ_{max} from 0.5 to 4, where$n$ was set to 2,000. Note that we masked the z-scores only to show the single effects of varying σ_{max} values. We displayed MAEs for both data sets in Table 2.

Increasing randomness makes accuracy worse as seen from Table 2. With increasing σ_{max} values, MAEs become worse for both data sets. The results are better for MLM than Netflix. The reason for this phenomenon is that MLM is denser than Netflix. When we compare the corresponding outcomes in Tables 1 and 2, MAE changes from 0.7082 to 0.7257 for MLM when σ_{max} is 2. For Netflix, MAE changes from 0.7400 to 0.7780. The accuracy losses for smaller σ_{max} values becomes negligible as expected.

In the third set of experiments, we scrutinized how varying β_{max} values affect overall performance. Note that β_{max} is another privacy control parameters. It determines the number of filled cells. With increasing β_{max} values, we fill in more unrated cells with random noise. Hence, we hypothesized that accuracy might be affected. We fixed σ_{max} at 2 while varied β_{max} from $d_u/4$,$d_u/2$, $3d_u/4$, d_u, and $2d_u$, where d_u represents the density of the user u's ratings vector. We again used 2,000 train users for both data sets. Due to the filled cells, PT might be affected, too. Thus, we also computed PT values. We conducted our experiments using MATLAB on a computer, which is Intel Core i7 with 3.60 GHz CPU. We displayed empirical outcomes for both data sets in Table 3.

Filling randomly selected with noise data makes accuracy slightly worse for both data sets as seen from Table 3. For Netflix, accuracy slightly becomes worse with increasing β_{max} values. However,

Table 2: Accuracy with varying σ_{max} values.

σ_{max}	**0.5**	**1**	**2**	**4**
MLM	0.7085	0.7144	0.7257	0.7604
Netflix	0.7409	0.7503	0.7780	0.8576

Table 3: Overall performance with varying β_{max} values.

	n	$\mathbf{d_u/4}$	$\mathbf{d_u/2}$	$\mathbf{3d_u/4}$	$\mathbf{d_u}$	$\mathbf{2d_u}$
MAE	MLM	0.7216	0.7273	0.7470	0.7323	0.7396
	Netflix	0.7690	0.7637	0.7626	0.7617	0.7612
PT	MLM	0.0551	0.0585	0.0616	0.0673	0.0892
	Netflix	0.0459	0.0462	0.0495	0.0548	0.0721

Table 4: Accuracy with varying n values.

N	**125**	**250**	**500**	**1,000**	**2,000**
MLM	0.8285	0.7777	0.7483	0.7350	0.7323
Netflix	0.8893	0.8411	0.8039	0.7745	0.7617

Table 5: MAEs with varying n when deviations are masked.

n	**125**	**250**	**500**	**1,000**	**2,000**
MLM	0.7638	0.7410	0.7240	0.7179	0.7154
Netflix	0.8011	0.7847	0.7689	07548	0.7528

varying β_{max} values cause almost the same accuracy losses because Netflix is a very sparse data set. Accuracy losses become slightly larger with increasing β_{max} values up to some point for MLM. As expected, it takes more time to estimate predictions with increasing β_{max} values for both data sets. Since more data are involved in prediction estimations with increasing β_{max} values, it is expected that PT values become larger. However, they are still smaller for both data sets even though the PT values are 0.0171 ms and 0.0139 ms for MLM and Netflix, respectively, in non-private case.

Another control parameter is n. The values of n might affect the performance. We hypothesized that increasing n values might improve accuracy. Effects of random numbers with increasing n values become smaller because average deviations are computed by dividing sum of the deviations by n. We fixed σ_{max} and β_{max} at 2 and d_u, respectively, while varied$n$ from 125 to 2,000 only. We displayed the outcomes in Table 4.

As seen from Table 4, the results displayed in the table verify our hypothesis. With increasing n values, accuracy becomes better for both data sets. Accuracy improvements become stable for larger n values. Improvements are larger when we increase n from 125 to 250 and from 250 to 500. As given in eqn (8), denominator becomes larger with increasing n values that smooth the effects of random noises. Hence, as expected, accuracy improves with increasing n values.

Users including the active users perturb their z-scores and send them to the server. However, as suggested by Basu *et al*. [11], each user can compute deviations and disguise them. We also studied how this approach affects our results. Thus, after computing z-scores and related deviations on z-scores, we disguised the deviations only without filling in any unrated cells. In this set of trials, we changed n from 125 to 2,000 only and fixed σ_{max} to 2. We displayed the results in Table 5 for both data sets.

As seen from Table 5, accuracy losses due to disguising deviations only are negligible. It is still possible to offer predictions with decent accuracy. As expected, the results become better with increasing n values. The effects of random noise become smaller because sum of the random noise is divided by n, as seen from eqn (8). For both data sets, the results improve with augmenting n values. Again, better outcomes are observed for MLM due to its higher density. Compared to the results for disguising z-scores, perturbing deviations only provides improved outcomes.

6 CONCLUSIONS AND FUTURE WORKS

Privacy is receiving increasing attention in collaborative filtering systems. Providing recommendations while preserving privacy is important. Weighted Slope One predictor is a successful collaborative filtering scheme. However, it is imperative to estimate predictions with privacy. In this paper, we first propose a normalized ratings (z-scores) based weighted Slope One predictor as a collaborative

filtering method. Since normalization usually enhances accuracy in similarity-based applications, we proposed a modified weighted Slope One predictor on z-scores. Our empirical outcomes show that our modified scheme performs better than the weighted Slope One predictor. We then used RPTs to disguise private data of both training and active users to achieve privacy. Our results show that it is still possible to estimate predictions with privacy. Accuracy losses are negligible due to randomization. Disguising deviations only provides better results due to smaller randomness.

As a future work, we are planning to scrutinize how to provide top-*N* recommendations using weighted Slope One predictor. We are also going to study how to estimate top-*N* recommendations with privacy.

ACKNOWLEDGEMENTS

This work is supported by TUBITAK under Grant 114E571.

REFERENCES

[1] Goldberg, D., Nichols, D., Oki, B.M. & Terry, D., Using collaborative filtering to weave an information Tapestry. *Communications of the ACM*, **35**(12), pp. 61–70, 1992. http://dx.doi.org/10.1145/138859.138867

[2] Bilge, A., Kaleli, C., Yakut, I., Gunes, I. & Polat, H., A survey of privacy-preserving collaborative filtering schemes. *International Journal of Software Engineering and Knowledge Engineering*, **23**(8), pp. 1085–1108, 2013. http://dx.doi.org/10.1142/S0218194013500320

[3] Park, D.H., Kim, H.K., Choi, Y. & Kim, J.K., A literature review and classification of recommender systems research. *Expert Systems with Applications*, **39**(11), pp. 10059–10072, 2012. http://dx.doi.org/10.1016/j.eswa.2012.02.038

[4] Ozturk, A. & Polat, H., From existing trends to future trends in privacy-preserving collaborative filtering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **5**(6), pp. 276–291, 2015. http://dx.doi.org/10.1002/widm.1163

[5] Canny, J., Collaborative filtering with privacy. *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, USA, pp. 45–57, 2002. http://dx.doi.org/10.1109/secpri.2002.1004361

[6] Canny, J., Collaborative filtering with privacy via factor analysis. *Proceedings of the 25th Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, pp. 238–245, 2002. http://dx.doi.org/10.1145/564376.564419

[7] Polat, H. & Du, W., Privacy-preserving collaborative filtering using randomized perturbationtechniques. *Proceedings of the 3rd IEEE International Conference on Data Mining*, Melbourne, FL,USA, pp. 625–639, 2003.

[8] Polat, H. & Du, W., Privacy-preserving collaborative filtering. *International Journal of Electronic Commerce*,**9**(4), pp. 9–35, 2005. http://dx.doi.org/10.1109/ICDM.2003.1250993

[9] Bobadilla, J., Ortega, F., Hernando, A. & Gutiérrez, A., Recommender systems survey. *Knowledge-Based Systems*, **46**, pp. 109–132, 2013. http://dx.doi.org/10.1016/j.knosys.2013.03.012

[10] Lemire, D. &Maclachlan, A., Slope one predictors for online rating-based collaborative filtering. *Proceedings of the SIAM Data Mining*, Newport Beach, CA, USA, pp. 471–475, 2005. http://dx.doi.org/10.1137/1.9781611972757.43

[11] Basu, A., Vaidya, J. & Kikuchi, H., Perturbation based privacy preserving Slope One predictors for collaborative filtering. *IFIP Advances in Information and Communication Technology*, **374**, pp. 17–35, 2012.
http://dx.doi.org/10.1007/978-3-642-29852-3_2

[12] Polat, H. & Du, W.,Effects of inconsistently masked data using RPT on CFwith privacy. *Proceedings of the ACM Symposium on Applied Computing*, Seoul, Korea, pp. 649–653, 2007.

[13] Yakut, I. &Polat, H., Achieving private SVD-based recommendations on inconsistentlymasked data. *Proceedings of the 1st International Conference on Security of Information and Networks*, Gazimagusa,North Cyprus, pp. 172–176, 2007.

[14] Bilge, A. &Polat, H., An improved privacy-preserving DWT-based collaborative filteringscheme. *Expert Systems with Applications*, **39**(3), pp. 3841–3854, 2012.
http://dx.doi.org/10.1016/j.eswa.2011.09.094

[15] Renckes, S., Polat, H. & Oysal, Y.,A new hybrid recommendation algorithm withprivacy. *Expert Systems*, **29**(1), pp. 39–55, 2012.

[16] Chow, R., Pathak, M.A. & Wang, C., A practical system for privacy-preserving collaborative filtering. *Proceedings of the International Workshop on Privacy in Social Data*, Brussels, Belgium, pp. 547–554, 2012.
http://dx.doi.org/10.1109/icdmw.2012.84

[17] Bilge, A. &Polat, H., A scalable privacy-preserving recommendation scheme via bisecting *k*-means clustering. *Information Processing & Management*, **49**(4), pp. 912–927, 2013.
http://dx.doi.org/10.1016/j.ipm.2013.02.004

[18] Basu, A., Kikuchi, H. & Vaidya, J.,Privacy-preserving weighted Slope One predictor for item-based collaborative filtering. *Proceedings of the International Workshop on Trust and Privacy in Distributed Information Processing*, Copenhagen, Denmark, 2011.

[19] Basu, A., Vaidya, J. & Kikuchi, H., Efficient privacy-preserving collaborative filtering based on the weighted Slope One predictor. *Journal of Internet Services and Information Security*, **1**(4), pp. 26–46, 2011.

[20] Gambs, S. &Lolive, J., SlopPy: Slope One with privacy. *Lecture Notes in Computer Science*, **7731**, pp. 104–117, 2013.
http://dx.doi.org/10.1007/978-3-642-35890-6_8

[21] Herlocker, J.L., Konstan, J.A., Borchers, A. & Riedl, J.T., An algorithmic framework for performing collaborative filtering. *Proceedings of the 22ndAnnual International ACMSIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, pp. 230–237, 1999.
http://dx.doi.org/10.1145/312624.312682