DynaEgo: Privacy-Preserving Collaborative Filtering Recommender System Based on Social-Aware Differential Privacy

Shen Yan[1,2,3], Shiran Pan[1,2,3], Wen-Tao Zhu[1,2(✉)], and Keke Chen[4]

[1] State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{yanshen,panshiran}@iie.ac.cn, wtzhu@ieee.org
[2] Data Assurance and Communication Security Research Center,
Chinese Academy of Sciences, Beijing, China
[3] University of Chinese Academy of Sciences, Beijing, China
[4] Data Intensive Analysis and Computing (DIAC) Lab, Kno.e.sis Center,
Wright State University, Dayton, OH, USA
keke.chen@wright.edu

Abstract. Collaborative filtering plays an important role in online recommender systems, which provide personalized services to consumers by collecting and analyzing their rating histories. At the same time, such personalization may unfavorably incur privacy leakage, which has motivated the development of privacy-preserving collaborative filtering (PPCF) mechanisms. Most previous research efforts more or less impair the quality of recommendation. In this paper, we propose a social-aware algorithm called DynaEgo to improve the performance of PPCF. DynaEgo utilizes the principle of differential privacy as well as the social relationships to adaptively modify users' rating histories to prevent exact user information from being leaked. Theoretical analysis is provided to validate our scheme. Experiments on a real data set also show that DynaEgo outperforms existent solutions in terms of both privacy protection and recommendation quality.

Keywords: Social networks · Privacy preserving · Recommender system · Collaborative filtering · Differential privacy

1 Introduction

Recommender systems [1] are widely used in e-commerce and online social networks, suggesting products, movies, music, etc. to consumers. Recommender systems are designed to provide personalized suggestions on items to consumers. Collaborative filtering (CF) plays an important role in recommender systems, which makes recommendations by collecting suggestions from similar users (user-based) or similar items (item-based).

However, the tendency towards personalization has raised privacy concerns [2]. To address this problem, privacy-preserving collaborative filtering (PPCF)

K.-Y. Lam et al. (Eds.): ICICS 2016, LNCS 9977, pp. 347–357, 2016.
DOI: 10.1007/978-3-319-50011-9_27

methods have been proposed, most of which employ traditional approaches such as cryptography, obfuscation, and perturbation. These approaches either induce great computation cost or provide insufficient privacy protection, thus impairing the performance in practical applications. In 2006, Dwork [3] proposed the notion of differential privacy (DP), which provides a quantifiable measurement of privacy. McSherry and Mironov [4] first introduced DP into CF. They constructed a private covariance matrix to randomize each user's ratings. In the context of the recommender systems, DP ensures that no user would be able to guess whether certain items appear in other users' rating histories.

Unfortunately, former methods fail to resist the k nearest neighbors (kNN) attack, which was first proposed by Calandrino et al. [5]. In the kNN attack, the adversary can infer the rating history of the target user by creating fake users. The success of the attack is due to that CF reveals users' preferences explicitly or implicitly, so the attacker can take advantage of the preferences to infer more information. Existent PPCF schemes only obscure the values of ratings, but not protect users' preferences. To address the problem, Guerraoui et al. [6] proposed D2P, which strengthens the notion of DP in recommender systems by creating a perturbed rating history named *AlterEgo* for each user. The perturbation process of D2P is highly dependent on subjectively selected parameters, which makes the privacy level non-intuitive and hard to control.

To improve both the privacy protection and the recommendation quality of former solutions, we introduce social relationships to PPCF schemes. Although social relationships have been widely used in recommender systems to boost the accuracy of recommendation, little work adopts it to provide effective privacy protection. In practice, users tend to share similar tastes with their close friends. So, substituting users's ratings with their friends' rating histories will obscure the exact ratings while keeping high utility. The import of social relationship information calls for us to customize the perturbed rating histories for different users. Therefore, the modified rating histories in our scheme are generated dynamically, which vary with the user who is receiving recommendations. We call our dynamically modified rating history *DyEgo*, and correspondingly name our scheme DynaEgo.

The rest of the paper is organized as follows. Section 2 introduces the preliminaries. In Sect. 3, we present our construction along with the theoretical analysis on privacy preserving. Section 4 reports the experimental results on a real data set called Epinions. Section 5 concludes this paper.

2 Preliminaries

2.1 Notation

Let $U = \{u_1, u_2, ..., u_n\}$ be a set of users and $I = \{t_1, t_2, ..., t_m\}$ be a set of items. The rating histories data set D can be represented as an $n \times m$ matrix, which can be decomposed into row vectors P_i and column vectors R_i. Specifically, the row vector $P_i = [r_{i1}, r_{i2}, ..., r_{im}]$ corresponds to the user u_i's rating history, and r_{ij} denotes the rating that user u_i gave to item t_j. The column vector

$R_i = [r_{1i}, r_{2i}, ..., r_{ni}]$ is composed of the ratings that t_i has received. For an item t_j that has not been rated by u_i, $r_{ij} = 0$. $sim(u_i, u_j)$ represents the similarity between users, and $itemsim(t_i, t_j)$ is the similarity between items. Empirical analysis indicates that Pearson correlation coefficient (Eq. 1) and Cosine similarity (Eq. 2) outperform other measurements, respectively:

$$sim(u_i, u_j) = \frac{\sum_{k=1}^{m} (P_i^{(k)} - \bar{P}_i)(P_j^{(k)} - \bar{P}_j)}{\sqrt{\sum_{k=1}^{m} (P_i^{(k)} - \bar{P}_i)^2 \sum_{k=1}^{m} (P_j^{(k)} - \bar{P}_j)^2}}, \tag{1}$$

$$itemsim(t_i, t_j) = \frac{\sum_{k=1}^{n} R_i^{(k)} \cdot R_j^{(k)}}{\sqrt{\sum_{k=1}^{n} R_i^{(k)} \sum_{k=1}^{n} R_j^{(k)}}}. \tag{2}$$

2.2 Differential Privacy

Definition 1 (ϵ-Differential Privacy). *A random function M satisfies ϵ- differential privacy if for all neighboring data sets D and D', and for all outputs t of this randomized function, the following statement holds:*

$$\frac{\Pr(M(D) = t)}{\Pr(M(D') = t)} \leq \exp(\epsilon), \text{ where } \exp(\epsilon) \triangleq e^{\epsilon},$$

Two data sets D and D' are said to be neighbors if they are different in at most one item. ϵ is the privacy protection parameter which controls the amount of distinction induced by two neighboring data sets. A smaller value of ϵ ensures a stronger privacy guarantee.

Two main mechanisms are usually used to achieve DP: the *Laplace* mechanism [7] and the *Exponential* mechanism [8]. The *Laplace* mechanism is only suitable for numeric outputs, while the *Exponential* mechanism is designed to address the privacy issues in scenarios of entity outputs.

Definition 2 (Exponential Mechanism). *Let $q(r)$ be a score function of data set D that measures the score of output $r \in D$. The mechanism M satisfies ϵ-differential privacy if $M(D)$ outputs r with the probability:*

$$\Pr(M(D) = r) = \frac{\exp(\epsilon q(r)/(2\Delta q))}{\sum_{r \in D} \exp(\epsilon q(r)/(2\Delta q))},$$

where Δq is the global sensitivity of q.

Definition 3 (Global Sensitivity). *The global sensitivity Δq of a function q is the maximum absolute difference obtained on the output over all neighboring data sets:*

$$\Delta q = \max_{D \sim D'} |q(D) - q(D')|.$$

Definition 4 (Composition Property [8]). *The sequential application of mechanisms M_i, each giving ϵ_i-differential privacy, gives $\sum \epsilon_i$-differential privacy.*

2.3 *k*NN Attack

The kNN attack assumes that the attacker knows m elements in P_t (i.e., partial rating history of u_t), which may be obtained by methods like social engineering. Then the attacker wants to infer the remaining items that u_t has rated.

The process of the kNN attack can be summarized as follows. The attacker creates k fake users (i.e., sybils), and assigns the partial rating history of u_t to each sybil's rating history. Then, with high probability, the k most similar users of each sybil will consist of u_t and the other $k-1$ sybils. The attacker inspects the list of items recommended to any of the sybils. Any item which appears in the list must have been rated by u_t.

2.4 Related Work

Former DP based PPCF solutions utilized DP mechanisms in various ways. McSherry and Mironov [4] proposed the first DP-based PPCF mechanism, which adds *Laplace* noise to the covariance matrix. PNCF [9] adopted the *Exponential* mechanism to select the neighbors, and add *Laplace* noise to the computation results. Zhu and Sun [10] proposed a mechanism based on the *Exponential* mechanism and designed a low sensitivity metric to measure the similarity.

Another method to adopt DP into PPCF is creating perturbed rating histories, e.g., *AlterEgo* in D2P [6]. In D2P scheme, the *AlterEgo* remains unchanged once created. In addition, the perturbation mechanism of D2P greatly depends on subjectively chosen parameters. Different from D2P, our proposal DynaEgo depends on the *Exponential* mechanism of DP, and employs social relationships to dynamically modify the rating histories.

3 DynaEgo Recommender

3.1 System Model

In this paper, we consider a general user-based CF recommender system, consisting of two parties: the service provider and the users. The service provider is trusted, storing users' rating histories and the social relationships data. There may be some malicious users in the system, who aim to infer other users' rating histories. The malicious users conduct the kNN attack. u_a is the active user, who is receiving recommendations. The service provider leverages k most similar users' rating histories to suggest items to u_a.

As Fig. 1 shows, DynaEgo relies on the *DyEgos* to make recommendations. The *DyEgos* are adaptively imitational rating histories. DynaEgo operates in three phases: Grouping phase, Modification phase, and Recommendation phase. The Recommendation phase is the same as the traditional CF algorithm, except using the *DyEgos* rather than the exact rating histories. Thus we only introduce the first two phases in detail below.

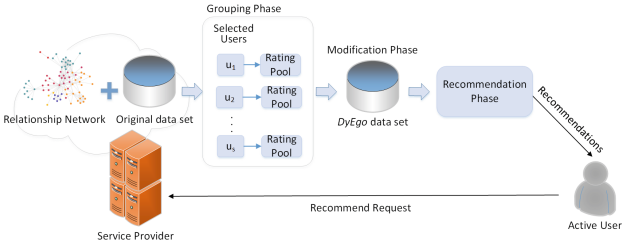

Fig. 1. The DynaEgo recommender scheme.

3.2 Grouping Phase

The Grouping phase consists of two steps. We select s users as the $SelectedUsers$ at first, then create a $RatingPool$ for each user in $SelectedUsers$.

$SelectedUsers$ are s users who have the most similar rating histories to u_a, where $s > k$ (k is the size of neighbors). To reduce the computation cost, only the rating histories of the $SelectedUsers$ will be substituted. Then, we create a $RatingPool$ for each u_i in $SelectedUsers$. We use the $Exponential$ mechanism of DP to randomly select h users for u_i, and add their ratings into the $RatingPool$ of u_i. More specifically, $sim(\cdot)$ acts as the score function, and $\Delta sim = 1$.

Algorithm 1 illustrates the process of creating the $RatingPool$ for a user u_t in $SelectedUsers$.

Algorithm 1. Grouping Function

Input: u_t, U, h, I
Output: $RatingPool_t$
  $UserPool = \{\}$
  for $i = 1 : h$ **do**
    Sample a user $u \in U$ with probability: $\frac{\exp(\epsilon_1 \cdot sim(u,u_t)/2)}{\sum_{u \in U} \exp(\epsilon_1 \cdot sim(u,u_t)/2)}$
    $UserPool.add(u)$, $U = U \backslash u$
  end for
  for u_i in $UserPool$ **do**
    for t_j in I **do**
      if $r_{ij} \neq 0$ **then**
        $RatingPool_t.add(r_{ij})$
      end if
    end for
  end for

3.3 Modification Phase

Our scheme relies on the $RatingPools$ obtained from the Grouping phase to create $DyEgos$ by the $Exponential$ mechanism. For users not in $SelectedUsers$, their $DyEgos$ are their real rating histories. Algorithm 2 shows the process of

Algorithm 2. Modification Function

Input: P_t, α, $RatingPool_t$
Output: $DyEgo_t$
$q(r_{ij}, r_{xy}) = \alpha \cdot trust(u_i, u_x) + (1-\alpha) \cdot itemsim(t_j, t_y)$
for r_{ij} in P_t **do**
Sample a rating $r_{xy} \in RatingPool_t$ with probability:

$$\frac{\exp(\epsilon_2 \cdot q(r_{ij}, r_{xy})/2)}{\sum_{r_{xy} \in RatingPool_t} \exp(\epsilon_2 \cdot q(r_{ij}, r_{xy})/2)}$$

$DyEgo_t.add(r_{xy})$
end for

creating $DyEgo$ for u_t. The social information is used to design the score function. It is believed that users with more common friends are more likely to have similar tastes. So we use the percentage of common friends (Eq. 3) to measure the trust relationship.

$$trust(u_i, u_j) = \frac{|F(u_i) \cap F(u_j)|}{|F(u_i) \cup F(u_j)|}. \tag{3}$$

Here, function $F(\cdot)$ returns the friend list of the user. $|F(u_i) \cap F(u_j)|$ is the number of common friends of u_i and u_j, and $|F(u_i) \cup F(u_j)|$ is the number of total friends of u_i and u_j.

According to Algorithm 2, we set the score function as $\alpha \cdot trust(u_i, u_y) + (1-\alpha) \cdot itemsim(t_j, t_x)$, where $\alpha \in (0, 1)$ is the parameter indicating the weight of social information in perturbations.

3.4 Theoretical Analysis

Theorem 1. *The recommendation mechanism M based on DynaEgo satisfies $(\epsilon_1+\epsilon_2)$-differential privacy, where ϵ_1 and ϵ_2 are privacy parameters in Grouping phase and Modification phase, respectively.*

First, we present lemmas and corollaries needed to prove Theorem 1.

Lemma 1. *We denote $SUB(i, j)$ the event of substituting element i with j in a user's rating history. For three arbitrary elements i, j, and k, where $i \neq j$, the following inequality holds:*

$$\frac{\Pr(SUB(i,k))}{\Pr(SUB(j,k))} \leq \exp(\epsilon_2),$$

where ϵ_2 is the privacy parameter in Modification phase.

The correctness of Lemma 1 can be easily derived from Definition 2 and Algorithm 2.

Lemma 2. *Let P_i denote the original rating history of user u_i, whereas P_i' and P_i are neighbors. PS is a privacy preserving mechanism which creates the DyEgo P_{iE} of u_i. Then, we have:*

$$\frac{\Pr(PS(P_i, P_{iE}))}{\Pr(PS(P_i', P_{iE}))} \leq \exp(\epsilon_2).$$

Proof. Let $P_i = [r_{i1}, r_{i2}, ..., r_{ik}, ..., r_{im}]$. Without loss of generality, we assume that P_i and P_i' only differ in the first element, i.e., $P_i' = [r_{i1}', r_{i2}, ..., r_{ik}, ..., r_{im}]$. r_{ik}^{π} denotes the permutation of the rating r_{ik}.

Based on the fact that every element is replaced independently, we get:

$$\Pr(PS(P_i, P_{iE})) = \prod_{k=1}^{m} \Pr(SUB(r_{ik}, r_{ik}^{\pi})), \tag{4}$$

$$\Pr(PS(P_i', P_{iE})) = \Pr(SUB(r_{i1}', r_{i1}^{\pi})) \cdot \prod_{k=2}^{m} \Pr(SUB(r_{ik}, r_{ik}^{\pi})). \tag{5}$$

Now, from Eqs. 4 and 5, we have:

$$\frac{\Pr(PS(P_i, P_{iE}))}{\Pr(PS(P_i', P_{iE}))} = \frac{\Pr(SUB(r_{i1}, r_{i1}^{\pi})) \cdot \prod_{k=2}^{m} \Pr(SUB(r_{ik}, r_{ik}^{\pi}))}{\Pr(SUB(r_{i1}', r_{i1}^{\pi})) \cdot \prod_{k=2}^{m} \Pr(SUB(r_{ik}, r_{ik}^{\pi}))}.$$

So, according to Lemma 1, we can get:

$$\begin{aligned}\frac{\Pr(PS(P_i, P_{iE}))}{\Pr(PS(P_i', P_{iE}))} &= \frac{\Pr(SUB(r_{i1}, r_{i1}^{\pi})) \cdot \prod_{k=2}^{m} \Pr(SUB(r_{ik}, r_{ik}^{\pi}))}{\Pr(SUB(r_{i1}', r_{i1}^{\pi})) \cdot \prod_{k=2}^{m} \Pr(SUB(r_{ik}, r_{ik}^{\pi}))} \\ \leq \frac{\exp(\epsilon_2 \Pr(SUB(r_{i1}', r_{i1}^{\pi})) \cdot \prod_{k=2}^{m} \Pr(SUB(r_{ik}, r_{ik}^{\pi}))}{\Pr(SUB(r_{i1}', r_{i1}^{\pi})) \cdot \prod_{k=2}^{m} \Pr(SUB(r_{ik}, r_{ik}^{\pi}))} &= \exp(\epsilon_2).\end{aligned}$$

□

Corollary 1. *For any two neighboring original rating histories data sets D, D', and an arbitrary DyEgo data set D^r, we denote $DS(D, D^r)$ as the function of substituting D^r for D. Then, we have:*

$$\frac{\Pr(DS(D, D^r))}{\Pr(DS(D', D^r))} \leq \exp(\epsilon_1 + \epsilon_2).$$

Proof. The process of building *DyEgo* set is composed of two phases: Creating the *RatingPool* and Modification phase. According to Lemma 2, the Modification phase satisfies ϵ_2-differential privacy. If the process of creating the *RatingPool* satisfies ϵ_1-differential privacy, the construction of *DyEgo* data set satisfies $(\epsilon_1 + \epsilon_2)$-differential privacy due to the composition property of DP. □

Proof of Theorem 1. Let M the mechanism based on DynaEgo, and M' is an arbitrary recommendation algorithm. We denote *Rec* as the recommendations given to u_a. So, Theorem 1 can be rewritten as:

$$\frac{\Pr(M(D, u_a) = Rec)}{\Pr(M(D', u_a) = Rec)} \leq \exp(\epsilon_1 + \epsilon_2).$$

In addition, we have:

$$\Pr(M(D,u_a)=Rec) = \sum\nolimits_{D^r} \Pr(DS(D,D^r))\Pr(M'(D^r,u_a)=Rec), \quad (6)$$

$$\Pr(M(D',u_a)=Rec) = \sum\nolimits_{D^r} \Pr(DS(D',D^r))\Pr(M'(D^r,u_a)=Rec). \quad (7)$$

From Eqs. 6, 7, and Corollary 1, we can get:

$$\frac{\Pr(M(D,u_a)=Rec)}{\Pr(M(D',u_a)=Rec)} = \frac{\sum_{D^r}\Pr(DS(D,D^r))\Pr(M'(D^r,u_a)=Rec)}{\sum_{D^r}\Pr(DS(D',D^r))\Pr(M'(D^r,u_a)=Rec)}$$
$$\leq \frac{\sum_{D^r}\exp(\epsilon_1+\epsilon_2)\Pr(DS(D',D^r))\Pr(M'(D^r,u_a)=Rec)}{\sum_{D^r}\Pr(DS(D',D^r))\Pr(M'(D^r,u_a)=Rec)} = \exp(\epsilon_1+\epsilon_2).$$

Therefore, we can conclude that M satisfies $(\epsilon_1+\epsilon_2)$-differential privacy. □

4 Experimental Evaluation

4.1 Setup

Data Set. We evaluate DynaEgo with the Epinions[1] data set. The Epinions data set consists of ratings given by 49,290 users over 139,738 items, and each rating ranges from 1 to 5. In addition, Epinions provides the social relationship data. In this paper, we use a subset of the Epinions data set with ratings given by 100 users over 500 items and the corresponding social relationship data, which shares the same size with the data set in [6].

Parameter Selection. The performance of PPCF varies with the privacy parameter ϵ. In our experiments, we set $\epsilon = 2$. According to [6], in D2P, ϵ is decided by three parameters: λ, p, and p^*. We set $\lambda = 0.5$, $p = 0.8$, and $p^* = 0.01$, which ensure that $\epsilon = 2$. Whereas in PNCF, the parameter ϵ can be selected directly. In our scheme, the privacy parameters ϵ_1 and ϵ_2 in Grouping phase and Modification phase are 1, ensuring the overall mechanism satisfies 2-differential privacy. In addition, without loss of generality, we set the size of $SelectedUsers$ twice as large as the neighbors' size, i.e., $s = 2k$. $h = 10$, and $\alpha = 0.2$.

4.2 Utility Evaluation

This section examines the utility performance of DynaEgo. We conduct experiments by defining different size of neighbors. We apply the traditional user-based CF as the baseline, and then compare DynaEgo with D2P and PNCF.

We measure the recommendation quality in terms of classification accuracy metrics (CAM) [11]. CAM measures the frequency with which a recommender makes correct or incorrect decisions about whether an item should be recommended. In CAM evaluation, parts of u_a's ratings are deleted, and we observe

[1] http://www.trustlet.org/epinions.html.

how well the recommendations match the real rating record. The recommender system selects the top-5 items as the recommendations. We compare the F_1-*Score* of the PPCF schemes and the non-private CF. The experiment for each mechanism is repeated for 50 times to eliminate the impact of randomization.

According to Fig. 2, DynaEgo maintains the F_1-*Score* larger than 0.6, which indicates that DynaEgo maintain a high recommendation quality. Specifically, when $\#neighbor = 15$, DynaEgo achieves an F_1-*Score* of 0.628, which outperforms the results of D2P (0.462) and PNCF (0.508) by 36% and 24%, respectively.

4.3 Privacy Evaluation

We use the *SuccessRate* of the kNN attack as the metric, which indicates the percentage of inferences that are correct (i.e., the items that u_t has rated).

We consider a recommender system with the configuration of 10 neighbors. An attacker creates 10 sybils, each of whom owns a rating history with parts of u_t's rating history (i.e., the auxiliary knowledge). Ideally, each sybil has a neighborhood consisting of u_t plus 9 other sybils. However, in PPCF schemes, the sybil cannot obtain neighbors as expected, due to the introduced perturbation.

In Fig. 3, each point represents the average value of *SuccessRate* from 50 repeated experiments. It may seem strange at the first sight that the *SuccessRate* decreases with the increase of auxiliary information. In fact, in PPCF schemes, more auxiliary knowledge increases the probability of other sybils rather than the target user appearing in the neighbors. Since sybil user's rating history does not have any information about u_t's remaining rating history, the increase of sybil users in neighbors has a negative effect on the inferences.

For the non-private scheme, the *SuccessRate* of kNN attack is 1.0, which indicates that the kNN attack always succeed. In PPCF schemes, the *SuccessRate* falls significantly. Compared with other PPCF schemes, DynaEgo has lower *SuccessRate* regardless the percentage of auxiliary knowledge of the attacker.

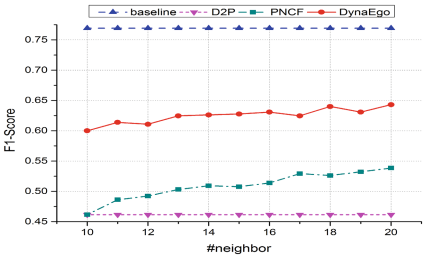

Fig. 2. Utility comparison between D2P, PNCF, and DynaEgo

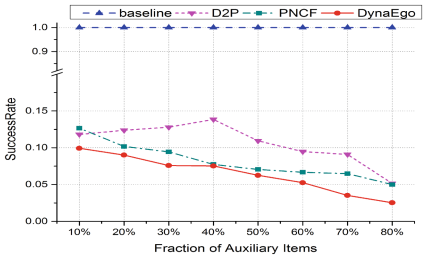

Fig. 3. Resistance of D2P, PNCF, and DynaEgo to the kNN attack.

4.4 Effect of Social Relationship Weight (α)

We vary the value of parameter α from 0.1 to 0.9 to observe the change of F_1-*Score* and *SuccessRate*, with the configuration of 80% auxiliary items and 15 neighbors. The larger α is, the bigger impact the social relationship will have.

Table 1. Effect of α on privacy and utility

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
F_1-*Score*	0.625385	0.622308	0.613846	0.612308	**0.607692**	0.618461	0.621026	0.625385	0.627692
SuccessRate	0.024365	0.021989	0.022166	0.019447	**0.017265**	0.021544	0.022121	0.021825	0.024005

Table 1 indicates that when $\alpha = 0.5$, our scheme will have the best resistance to the kNN attack and a correspondingly lower F_1-*Score* value. The privacy protection degrades when α approaches 0.1 or 0.9.

5 Conclusions

In this paper, we have proposed a social-aware PPCF scheme called DynaEgo. The main idea of DynaEgo is substituting users' rating histories adaptively, which adopts the *Exponential* mechanism of DP. Additionally, social relationships are used to design the score function. To evaluate the performance of our scheme, we have conducted experiments on the Epinions data set. The results show that DynaEgo outperforms existent solutions in both utility and privacy protection. Since DynaEgo is independent of any specific recommendation algorithm, it can be applied to all recommender systems that are based on users' rating histories.

Acknowledgment. This work was supported by the National Natural Science Foundation of China under Grant 61272479, the National 973 Program of China under Grant 2013CB338001, and the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA06010702.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
2. Ramakrishnan, N., Keller, B.J., Mirza, B.J., Grama, A.Y., Karypis, G.: Privacy risks in recommender systems. IEEE Internet Comput. **5**(6), 54 (2001)
3. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). doi:10.1007/11787006_1
4. McSherry, F., Mironov, I.: Differentially private recommender systems: building privacy into the net. In: 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2009), pp. 627–636. ACM (2009)

5. Calandrino, J.A., Kilzer, A., Narayanan, A., Felten, E.W., Shmatikov, V.: You might also like: privacy risks of collaborative filtering. In: 32nd IEEE Symposium on Security and Privacy (S&P 2011), pp. 231–246. IEEE (2011)
6. Guerraoui, R., Kermarrec, A.M., Patra, R., Taziki, M.: D2P: distance-based differential privacy in recommenders. Proc. VLDB Endow. **8**(8), 862–873 (2015)
7. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). doi:10.1007/11681878_14
8. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), pp. 94–103. IEEE (2007)
9. Zhu, T., Li, G., Ren, Y., Zhou, W., Xiong, P.: Differential privacy for neighborhood-based collaborative filtering. In: 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013), pp. 752–759. ACM (2013)
10. Zhu, X., Sun, Y.: Differential privacy for collaborative filtering recommender algorithm. In: 2nd ACM International Workshop on Security and Privacy Analytics, pp. 9–16. ACM (2016)
11. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. (TOIS) **22**(1), 5–53 (2004)