

# Privacy-Enhanced Collaborative Filtering

Shlomo Berkovsky<sup>1</sup>, Yaniv Eytani<sup>1</sup>, Tsvi Kuflik<sup>2</sup>, Francesco Ricci<sup>3</sup>

<sup>1</sup> Computer Science Department, University of Haifa, Israel  
{slavax, ieytani}@cs.haifa.ac.il

<sup>2</sup> Management Information Systems Department, University of Haifa, Israel  
tsvikak@is.haifa.ac.il

<sup>3</sup> ITC-irst, Trento, Italy  
ricci@itc.itc

**Abstract.** Current implementations of the Collaborative Filtering (CF) algorithm are mostly centralized and the information about users (their profiles) is stored in a single server. Centralized storage poses a severe privacy hazard, since user profiles are fully under the control of the recommendation service providers. These profiles are available to other users upon request and are transferred over the network. Recent works proposed to improve the scalability of CF by distributing the stored profiles between several repositories. In this work we investigate how a decentralized approach to users' profiles storage could mitigate some of the privacy concerns of CF. The privacy hazards are resolved by storing the users' profiles only on the client-side so they are used for computation similarity only on the client-side. Only a value indicating the similarity is transferred over the network, without revealing the profile itself. To further avoid the disclosure of the user's profile through a series of attacks, we propose that the users hide or obfuscate parts of their profile. Experimental results show that relatively large parts of the user's profile could be obfuscated without hampering the accuracy of the CF.

## 1 Introduction

Collaborative Filtering (CF) is commonly used in the E-Commerce realm for producing recommendation for various products. CF is based on the assumption that people with similar tastes prefer the same items. In order to generate a recommendation, CF initially creates a neighborhood of users with the highest similarity to the user whose preferences are to be predicted. Then, it generates a prediction by calculating a normalized and weighted average of the ratings of the users in the neighborhood.

In CF, user profile is a feature-vector containing information about user preferences with respect to a set of item the user rated. For quite some time CF has been applied in E-Commerce and direct recommendations of various kinds [15]. Personalized information delivery in general and purchase recommendations (that applies collaborative filtering) in particular can increase the likelihood of a customer making a purchase, compared to non-personalized approaches.

However, personalization brings with it the issue of privacy. Privacy is an important challenge facing the growth of Internet and the acceptance of various transaction models supported by Internet. Basically, Web users leave identifiable tracks while surfing the Web and there is a growing awareness and concern about the misuse of such information [1]. Many eavesdroppers on the Web violate users' privacy for their own commercial benefits, and as a result, users concerned about their privacy refrain from using useful Web services to prevent exposure [6], [4].

The need for protection of Web users' privacy triggers growing research efforts nowadays. Wide variety of research directions for preservation the privacy while surfing the Web in general, and for CF in particular are explored. Canny [2], [3] suggests privacy preservation approach based on peer-to peer techniques. He suggests forming users' communities, where the community will have an aggregate user profile, representing the group as whole and not individual users. Personal information will be encrypted and the communication will be between individual users and not servers. Thus, the recommendation will be generated on the client side.

Polat and Du [11], [12] suggest another method for preservation of user privacy on the central server by adding uncertainty to the data by using a randomized perturbation technique. Hence, the server (or the data collector) has no knowledge about true values of individual users rated items. They demonstrate that this method does not lower considerably the obtained accuracy of the results

In today's dynamic environments, formation of a community of users poses limitation on possibilities of information sharing. Users looking for personal information in various domains and situations may need to interact with different set of users every time. To accomplish this while preserving the users' privacy, we propose an approach that combines benefits from both Canny, and Polat and Du. Generally, we believe that users should control when and what personal information they would like to reveal.

Individual users may participate in a virtual, distributed CF system in the following way: every user may keep and maintain his personal profile of rated items. Recommendation is requested by a user sending parts of his profile and a request for recommendation. Other users may decide to respond to that request by sending their recommendation and degree of similarity with the requester. The originator collects the responses and uses them as a source for neighborhood formation that later on leads to local generation of recommendation. Such approach preserves users' privacy by leaving them in control of their personal information, while allowing them to support recommendation generation by other users.

We experimented with the publicly available Jester dataset containing rating for 100 jokes [5]. Results clearly demonstrate that adding the proposed privacy enhancements do not severely affect the accuracy of the recommendations obtained basing on the CF algorithm.

The rest of the paper is structured as follows: Section 2 discusses limitations of centralized CF and distributed CF. Section 3 presents our "on-the-fly" CF approach for recommendation generation. Section 4 presents the experimental results validating the approach and discusses open research questions. Section 5 concludes the work and presents directions of future research.

## 2 Distributed Collaborative Filtering and Privacy

Centralized CF systems have a number of disadvantages. In particular, they pose a severe threat to users' privacy, as the service providers collect valuable information about the users. This information can be transferred or sold once it was collected and used of malicious purposes. Thus, according to a recent survey [4], most users will not agree to divulge their private information. This causes users to refrain from providing personal information or to provide false information. Using CF without compromising user's privacy is most certainly an important and challenging issue.

[11] suggested a method for preservation of user privacy on the central server by adding uncertainty to the data. Before transferring personal data to the server, each user first "disguises" using a randomized perturbation technique. Therefore, the server (and also the attacker) can not find out the actual contents of users' profile. Although this method changes the user's original data, experiments show that the modified data still allows providing relatively accurate recommendations. This approach enhances users' privacy, but the users still depend on centralized, domain-specific servers that they subscribe to, on getting a recommendation. The capability to dynamically receive "on-the-fly" recommendations is limited.

In general, storing users' profiles on several locations reduces the risk of having the data exposed to an attacker in comparison to a storage on a single server. CF over a distributed setting of data repositories was initially proposed in [16]. This work presented a Peer-to-Peer architecture supporting product recommendations for mobile customers represented by software agents. The communication between the deployed agents used expensive routing mechanism based on network flooding that increased the communication overhead. An improved mechanism was proposed in [10], however it reduced the efficiency of neighborhood formation phase. The work in [14] elaborated the discussion on distributed CF. It developed a detailed taxonomy of distributed CF in recommender systems and presented different implementation frameworks for different domains of Electronic Commerce. PocketLens project [9] implemented and compared five distributed architectures for CF. It was found that no architecture is perfect, but the performance of content-addressable mechanism [13] is close to the performance of centralized CF algorithm.

Other approach to distributed recommendation is by completely eliminating the use of servers. A user creates a query by sending a part of his profile and a request for recommendation on this specific item. Other users decide to respond and send their information to the requester. However, this approach still requires transferring the users' profiles over the network, thus posing privacy issues. Two schemes for privacy-preserving CF were proposed in [2] and [3]. In these schemes the users control all of their own private data, while a community of users can compute a public "aggregate" of their data without disclosing individual users' data. The aggregate allows personalized recommendations to be computed by members of the community, or by outsiders. This approach protects users privacy in a distributed setting, but requires users group formation and complicated communication which are limitations in today's evolving dynamic environments

### 3 On the-fly recommendation generation

Users are the owners of their personal information. Thus, they should decide if, when and how to reveal parts of their user-profile. Recommendations may be requested in specific context – specific domain, in specific time and location. The users need to have continuous access to recommendation systems, not limited by availability of servers or predefined users groups.

The easiest way to support this request is by applying “Peer-to-Peer” recommendations (figure 1). Revealing a user profile is required in two different cases. The first is when a user requests a recommendation from other users. In this case the user must reveal his/her own profile in order to receive relevant information (recommendation). The second case is when a user decides to provide recommendation to other users by revealing his/her user profile, so the recommendation requester can use it in order to construct a neighborhood of similar users and generate a recommendation.

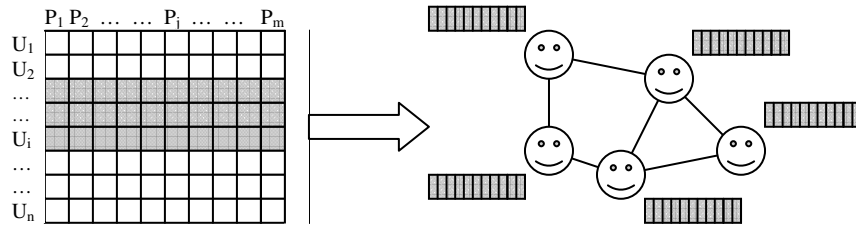


Fig. 1. Centralized vs. Decentralized Storage of Users' Profiles

In both cases users' privacy is at risk. We try to preserve user's privacy in the following way. Like Canny [2],[3], we believe that recommendations should be provided by individuals, at will. However, today's dynamic online environment prevents formation of communities and aggregation of users' profiles. For requesting recommendation, user may reveal only the relevant part of his profile, and even this part can be partially revealed. This solution is similar to the approach taken by Polat and Du in [11] and [12]. Only a subset of the features ratings should be provided. In addition, when providing a recommendation to another user, only a recommendation with a degree of similarity can be provided, instead of the actual user profile.

There are few questions that are posed when considering this approach. The main question is what portion of a user profile is needed in order to generate a recommendation. This is relevant for both sides – the recommendation requester and the recommendation provider. Another question is the question of trust – how trustworthy are the recommenders.

The current work addresses the first question and demonstrates that it is possible to use a relatively small portion of user profile in order to generate good recommendation. This is true for both the information requester and the responding users. The practical meaning is that users may protect their privacy simply by revealing small and partial portions of their profile when requesting information or providing a recommendation.

Though this approach improves the privacy of the responding users, it still allows to reveal the users' profile thought a systematic attack using similarity requests. We

further increase the users’ privacy by obfuscating parts of their profiles in the process of calculating the similarity. Thus, the values collected by an attacker will not be reliable enough for a single user and complicate the task of inferring about the real contents user profile. However, since the system has many users such local rating obfuscation should not hamper too much the overall system performances.

## 4 Experimental results and discussion

In the experimental part of our work we used Jester dataset of jokes. Jester is a web-based joke recommendation system, developed at Berkeley University [5]. The database contains 4.1 million continuous ratings (-10.00 to +10.00) of 100 jokes from 73,421 users. Significant part of the users end up reading and rating all the jokes, so Jester dataset is relatively dense. In total, almost 50% of all possible ratings in the matrix are present.

We chose a subset of 1,024 users that rated all 100 jokes to get a dense matrix where every rating in the matrix corresponds to an actual user rating. We simulated decentralized distributed environment by a Java multi-threaded implementation. Each request was transferred to the relevant users, each user computed the similarity locally (possibly on an obfuscated profile), and returned the similarity rate and the required rating to the originator of the request. Upon receiving the responses from the other users, the originator generates the prediction locally as a weighted average of the ratings of the most similar users. Hence, the process for generating the prediction is performed in the same manner as in a centralized CF, except the similarity computation that is done distributively.

The first experiment aims to test how storing profiles at the client-side influences the accuracy of the generated recommendations. To measure the accuracy of the prediction we used Mean Average Error (MAE) [8] metrics. MAE was computed by:

$$MAE = \frac{\sum_{i=1}^N |p_i - r_i|}{N},$$

where  $N$  denotes the total number of the generated predictions,  $p_i$  is the  $i^{th}$  prediction, and  $r_i$  is the real  $i^{th}$  rating.

The MAE results obtained in the experiment are relatively low, between 0.15 and 0.18, implying that generated predictions are similar to the real ratings of the originating users. These MAE values are also similar to ones obtained at previous works that used the Jester dataset (initially presented in [5], and recently compared in [3]). Despite the fact that a centralized storage was distributed between the users, the actual ratings in the profiles remained unchanged. Thus, there is no reason to expect different MAE values.

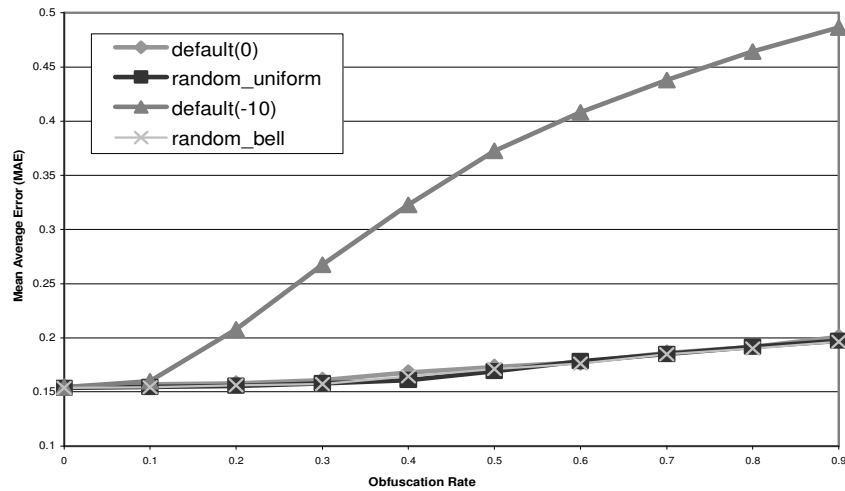
The second experiment aims at determining the influence of data obfuscation on the recommendation accuracy. We compared four different methods for modifying the data in users’ profiles. We measured the effect of gradually replacing increasing parts of the users’ profiles with either a predefined value or randomly chosen value. When replacing real values with predefined ones, we also tested the effect of the choosing various values.

Thus, we define three basic policies for modifying the data in users' profiles:

- Uniform Random obfuscation – real ratings values in the user's profile are substituted by a random values chosen uniformly in the scope of possible ratings  $(-10.00$  to  $+10.00)$ .
- Bell curved Random obfuscation – real ratings values in the user's profile are substituted by random values chosen using a bell curve distribution with similar statistics to the dataset.
- Default obfuscation( $x$ ) – real ratings values in the profile are substituted by a predefined value  $x$ .

To check the influence of  $x$  in obfuscation( $x$ ) policy on the accuracy of recommendation, we conducted this experiment with two different values of:  $x=0$  (which is close to the average of the ratings of the dataset), and  $x=-10$  (an extremely negative rating).

At each experiment we gradually increased the percentage of user profile that is modified (further referred as obfuscation rate) from  $0.0$  (the original profile is unchanged) to  $0.9$  (90% of the ratings in a profile of each user are modified). We produced a fixed testing set of  $10,000$  random jokes, and for each possible obfuscation rate we measured the MAE for the whole testing set. Figure 2 illustrates MAE results as a function of the obfuscation rate.



**Fig. 2.** MAE vs. Obfuscation Rate

Figure 2 shows that the performance of both Random policies and Default( $0$ ) obfuscation policies is similar. These policies do not drastically change the accuracy of the generated recommendations. The MAE rate slightly increases as the obfuscation rate increases, however the change is minor (from  $0.15$  to  $0.2$ ) and the prediction is still accurate. We explained it by considering that both the average rating value of the Default( $0$ ) obfuscation and the expectation rating value of the Uniform Random obfuscation is equal to  $0$ . The expectation rating value of the bell curved Random obfuscation is equal to  $1$ . These values are close to the average rating value of the ratings in the dataset. Thus, substituting the actual ratings with similar ratings creates only a small overall impact on the MAE computed over many users.

On the other hand, using the Default(-10) obfuscation policy, the actual ratings are substituted by a highly dissimilar value, (as it is far from the average value in the data set). As a result, the MAE rate increases linearly starting from 10% obfuscation rate.

#### 4.1 Discussion

The experimental results demonstrate that it is possible to use a relatively small portion of user profile in order to generate good recommendation. However, these results raise a number of interesting research questions related not only to privacy, but to other fundamental CF topics in general. In the rest of this section we briefly describe these questions:

- The experiments were performed on a dense subset of Jester dataset. Despite that, we claim that obfuscating part of the data has the same impact as working with sparse dataset.
  - Will the results change when conducting the same experiments on a sparse dataset, e.g., MovieLens?
  - Could a differential obfuscation policy be developed to minimize loss of real data in sparse datasets?
- Analyzing the nature of the data. In Jester dataset (and supposedly in other datasets) there are many items that most users agree on their rating,
  - Can random selection provide similar results to a sophisticated recommendation system?
- The obfuscated results show that all the users have basically very similar profiles and still the prediction is good. This means that most users tend to prefer the same jokes.
  - Could it be confirmed on another data set?
  - What about “non-standard” users that have different preference, how will obfuscation influence them?
- Standalone attacks of malicious users through changing their ratings can not affect the accuracy of the global predictions.
  - How will our approach scale under an organized attack of multiple users hamper the functionality of CF?
- Our approach still requires transferring the originator profile over the network posing privacy issues.
  - Can the originator’s profile be perturbed in a similar way?
- In a real situation only a fraction of the peers are likely to respond.
  - How many peers to will be needed to communicate with and ultimately what is the scalability and cost of the algorithm?

We believe that these questions require additional research before practical conclusions should be drawn regarding the contribution of the demonstrated approach.

## 5 Conclusions and Future Research

The need to protect of users' privacy triggers growing research efforts. Many eavesdroppers on the Web violate users' privacy and as a result users concerned about their privacy refrain from using useful Web services to prevent exposure. Additionally, in today's dynamic environments, formation of a community of users poses limitation on possibilities of information sharing. Users looking for personal information in various domains and situations may need to interact with different set of users every time.

This work proposes a simple and effective solution for preservation of the users' privacy during information sharing interaction. It employs a privacy-enhanced CF algorithm that allows creating dynamic and distributed recommendations. These recommendations are generated "on-the-fly" by letting the individual users participate in a virtual, distributed CF system. The users control when and what is the personal information they reveal.

Our approach stores users' profiles on several different locations and thus has the advantage of reducing the risk of having the users' data exposed to a malicious attacker. Moreover it can decrease the likelihood that the information could be collected for the purpose of transferring or selling and then be used in malicious way.

In order to further increase users' privacy, parts of the users' profiles are obfuscated in the process of calculating their similarity to the originator user. Thus, values collected by an attacker will not be reliable enough for a single user and complicate the task of inferring about the real contents of the user profile. As the system has many users such local obfuscations does not hamper the overall system performances.

### 5.1 Future Research

The current work demonstrated that it is possible to use a relatively small portion of user profile in order to generate good recommendation. However, in a real situation, only a fraction of the users are likely to respond. That means the user originating the request will have to send their data to a much larger set of peers. A natural extension of our approach would be to study the accuracy of privacy enhanced CF as a function of the number of peers who responded. This will answer an important question for determining how many peers to should be communicated with, and ultimately the scalability and cost of the algorithm.

Another future research direction is the problem extreme sparseness in the CF domains. Available movies databases (who have several thousand titles) are still at the low end of the number of choices, and are relatively dense. However, there are many more CDs or books, or TV shows to choose from (and in practice, these items follow a Zipf distribution). Thus, sparseness is an inescapable reality for most practical CF domains. Unfortunately, statistical obfuscation and sparseness do not correlate well together. Perturbing missing data items could swamps the information found in the real data. We plan to investigate how our approach behaves on real sparse datasets and possible ways to improve it.

## References

- [1] S.Brier, "How to Keep your Privacy: Battle Lines Get Clearer", The New York Times, 13-Jan-97.
- [2] J.Canny, "Collaborative Filtering with Privacy", in IEEE Symposium on Security and Privacy, Oakland, CA, 2002.
- [3] J.Canny, "Collaborative Filtering with Privacy via Factor Analysis", in proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, 2002.
- [4] L.F.Cranor, J.Reagle, M.S.Ackerman, "Beyond Concern: Understanding Net Users' Attitudes about Online Privacy", Technical report, AT&T Labs-Research, April 1999.
- [5] K.Goldberg, T.Roeder, D.Gupta, C.Perkins, "Eigentaste: A Constant Time Collaborative Filtering Algorithm", in Information Retrieval, 4(2), pp.133-151, 2001.
- [6] P.Harris, "It is Time for Rules in Wonderland", Businessweek 20, 2000.
- [7] J.Herlocker, J.A.Konstan, J.Riedl, "Explaining Collaborative Filtering Recommendations", in proceedings of ACM Conference on Computer Supported Cooperative Work, Philadelphia, PA, 2000.
- [8] J.L.Herlocker, J.A.Konstan, L.G.Terveen, J.T.Riedl, "Evaluating Collaborative Filtering Recommender Systems", in ACM Transactions on Information Systems, vol.22(1), pp.5-53, 2004.
- [9] B.N.Miller, J.A.Konstan, J.Riedl, "PocketLens: Toward a Personal Recommender System", in ACM Transactions on Information Systems, vol.22 (3), 2004.
- [10] T.Olsson, "Decentralised Social Filtering based on Trust", in proceedings of AAAI-98 Recommender Systems Workshop, Madison, WI, 1998.
- [11] H.Polat, W.Du, "Privacy-Preserving Collaborative Filtering Using Randomized Perturbation Techniques", in proceedings of International Conference on Data Mining, Melbourne, FL, 2003.
- [12] H.Polat, W.Du, "SVD-based Collaborative Filtering with Privacy", in proceedings of ACM Symposium on Applied Computing, Nicosia, Cyprus, 2004.
- [13] S.Ratnasamy, P.Francis, M.Handley, R.Karp, S.Shenker, "A Scalable Content-Addressable Network", in proceedings of ACM SIGCOMM, San Diego, CA, 2001.
- [14] B.M.Sarwar, J.A.Konstan, J.Riedl, "Distributed Recommender Systems: New Opportunities for Internet Commerce", a chapter in "Internet Commerce and Software Agents: Cases, Technologies and Opportunities", Idea Group Publishers, 2001.
- [15] J.B.Schafer, J.A.Konstan, J.Riedl, "E-Commerce Recommendation Applications", in Journal of Data Mining and Knowledge Discovery, vol. 5 (1/2), pp. 115-152, 2001.
- [16] A.Tveit, "Peer-to-Peer Based Recommendations for Mobile Commerce", in proceedings of the 1st International Workshop on Mobile Commerce, Rome, Italy, 2001.