Hierarchical Neighborhood Topology for Privacy Enhanced Collaborative Filtering

Shlomo Berkovsky
Computer Science Department,
University of Haifa, Israel
slavax@cs.haifa.ac.il

Yaniv Eytani
Computer Science Department,
University of Illinois at Urbana Champaign, USA
yeytani2@uiuc.edu

Tsvi Kuflik
Management Information Systems Department,
University of Haifa, Israel
tsvikak@is.haifa.ac.il

Francesco Ricci
ITC-irst,
Trento, Italy
ricci@itc.it

ABSTRACT
Privacy is an important challenge facing the growth of the Web and the propagation of various transaction models supported by it. Decentralized distributed models of computing are used to mitigate privacy breaches by eliminating a single point of failure. However, end-users can still be attacked in order to discover their private information. This work proposes using distributed hierarchical neighborhood formation in the CF algorithm to reduce this privacy hazard. It enables accurate CF recommendations, while allowing an attacker to learn at most the cumulative statistics of a large set of users. Our approach is evaluated on a number of widely-used CF datasets. Experimental results demonstrate that relatively large parts of the user profile can be obfuscated while a reasonable accuracy of the generated recommendations is still retained. Furthermore, only a small subset of users may be required for generating accurate recommendations, suggesting that the proposed approach is scalable.

Author Keywords
Recommender Systems, privacy, Collaborative Filtering

ACM Classification Keywords
Information Filtering, Decision Support, Machine Learning, Knowledge Personalization and Customization.

INTRODUCTION
Privacy is an important challenge facing the growth of the Internet and the acceptance by users of various transaction models which it supports. Personalized information delivery in general, and products recommendation in particular nowadays play a major role in the development of the Web [15]. Such methods can increase the likelihood of a customer making a purchase, compared to non-personalized approaches. However, Web users leave identifiable tracks while surfing the Web, and there is a growing awareness of and concern about the misuse of such information [1]. Many eavesdroppers on the Web violate user privacy for their own commercial benefits, and as a result, users concerned about their privacy refrain from using Web applications, just to prevent possible exposure [6]. According to a recent survey [4], most users will not agree to openly sharing their private information. However, people are not equally protective of every attribute in their data records [17, 4]. A user may not divulge the values of certain attributes at all, may not mind giving true values for others, or may be willing to share private information by giving modified values of certain attributes.

Privacy hazards for personalization system are aggravated by the fact that effective personalization requires large amounts of personal data. For example, collaborative filtering (CF), commonly used in the E-Commerce recommender systems [15], is based on the assumption that people with similar tastes expressed in the past will prefer similar items in the future. Here, a user's preferences are modeled as a vector containing his or her opinions on a set of items, expressed by explicit ratings provided by the user on these 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 (based on the similarity/correlation of their rating vector representations). Then, it generates a recommendation by averaging the ratings of the users in the neighborhood for the given item [16]. Clearly, the accuracy of the recommendations thus generated is correlated with the number of similar users and the degree of their similarity. The more detailed are the user profiles and the larger their cumulative number, the more reliable will be the recommendations. Hence, there is a clear trade-off between the accuracy of the provided personalization and the privacy of user data. As more personal data is revealed, better and more accurate recommendations are generated.

In real life scenarios, the ever growing amount of data about users and products (or items) may be naturally distributed among many data repositories. A single data repository may be focused around only a limited variety of topics or domains (e.g., movies or books). Hence, users

looking for accurate personalized information, possibly of various kinds, may well need to interact with a different set users and systems every time. Doing so can ensure that sufficient information relevant to the user's query is collected and that the recommendations produced are accurate. Distributed infrastructures can facilitate the development of such personalized environments. These can also be useful to mitigate some privacy breaches by eliminating a single point of failure. However, insecure end-user communication in distributed environments can be attacked and private information may still be exposed.

Hence, in order to provide a stable dynamic infrastructure while preserving the users' privacy, a previous study [2] suggested perturbing parts of the user's profiles [11] while using a decentralized distributed infrastructure [3]. This setting allows users to store their personal profile locally and leaves them in control as to what personal information they would like to reveal, and when. Thus, a user (hereafter referred to as the *active* user) requesting, for instance, similar user profiles for generating a CF recommendation, would receive only modified user profiles. From these profiles the active user can learn only a limited amount of information about the true ratings of individual users. Then, the active user aggregates ratings of most similar users to generate the recommendations locally. Experimental results indicated that this method does not lower considerably the obtained accuracy of the generated recommendations.

This work extends [2] by providing new both methodological and experimental contributions. First, we propose exploiting the notion of hierarchical topology (for example, see [18]). In this setting, peers are organized into peer-groups managed by *super-peers*. The super-peers encapsulate computations made by the underlying peers and then aggregate their results before sending them to the active user. Similarly, the active user aggregates the responses of the super-peers and generates a recommendation. In this scenario, attacking one of the super-peers does not yield any meaningful information about any individual user. An attacker may not learn the ratings of a single user, but only the average preferences of a large group of users (managed by the given super-peer) [12, 9]. To increase privacy, each super-peer chooses only a random subset of its peers to form the neighborhood of similar users. Within the peer-groups, privacy can be additionally preserved by using the previously introduced obfuscation methods [2] and through querying only a subset of peers. Thus, our approach preserves users' privacy by leaving them in control of their personal information, while allowing them to support recommendation generations initiated by other users.

Previous work has examined the issue of obfuscating user profiles only on a dense dataset (e.g., a subset of Jester's user base), where the intended meaning of the term "dense" is that a large percentage of all the possible user product ratings are available. In this paper we evaluate the obfuscation approach and the scalability of the proposed hierarchical extension using three publicly available datasets: Jester[5], MovieLens [7], and EachMovie [14]. Thus, we experimented with both dense (Jester) and very sparse datasets (MovieLens and EachMovie). In the latter only a small fraction of all possible ratings are known. Results for all datasets demonstrate that a relatively large part of the user profile can be obfuscated, and only a small subset of users is required to generate a recommendation without hampering the accuracy of the CF. Hence, adding the proposed privacy enhancements does not severely affect the accuracy of the CF recommendation algorithm, and it is scalable by the number of peers.

The rest of the paper is structured as follows: Section 2 presents the distributed hierarchical CF approach and discusses the obfuscation policies. Section *3* presents the experimental results validating the approach and discusses the empirical evidence. Section *4* concludes the work, and presents directions for future research.

DISTRIBUTED RECOMMENDATION GENERATION

Users looking for personalized information in various domains and situations may need to interact with sets of other users. During this process, they reveal their own profile and request recommendations from other users (or service providers). To provide personalization, while preserving user privacy, [2] suggests using a distributed infrastructure and obfuscating parts of the user profiles before sending them over the underlying communication middleware. Obfuscation reduces the amount of a user's information exposed to other users. It shows that it is possible to obfuscate relatively large portions of a user's profile, and still generate accurate recommendations. This observation is true for both the information requester and the responding users. Practically, this means that users may protect their privacy simply by revealing small portions of their profile when requesting personalization or providing recommendations to other users.

This approach enhances the privacy of the responding users. However, it still allows the responding users' profiles to be revealed through a systematic attack using multiple requests. In order to prevent this scenario, we propose the notion of peer groups and super-peers, where each group contains a dynamic number of users. When a request for a recommendation is received, one of the users (negotiated within the group every time) is elected as a "super-peer" and becomes a communication mediator between the requester of the information and the users in the underlying peer group. In this way each super-peer logically encapsulates the data of the underlying peers. Within peer-groups, privacy could be additionally enhanced using the obfuscation methods and querying a random subset of peers.

Consider the following example illustrating the above ideas. A recommender system keeps track of three peer groups. Upon receiving a request from an active user, the system forwards it to the peer groups. Within each peer group, an

ad-hoc selected super-peer handles the request. It selects a subset of the peers in the group and forwards the request only to them. Upon receiving the responses from the selected peers, each super-peer builds a *local* recommendation, and forwards it to the active user jointly with the number of users in the cluster and the average similarity of the *K* nearest neighbors in the local neighborhood. Upon collecting the results from all the super-peers, the active user generates a *global* recommendation by aggregating three obtained local recommendations. The above process is schematically depicted in Figure 1.

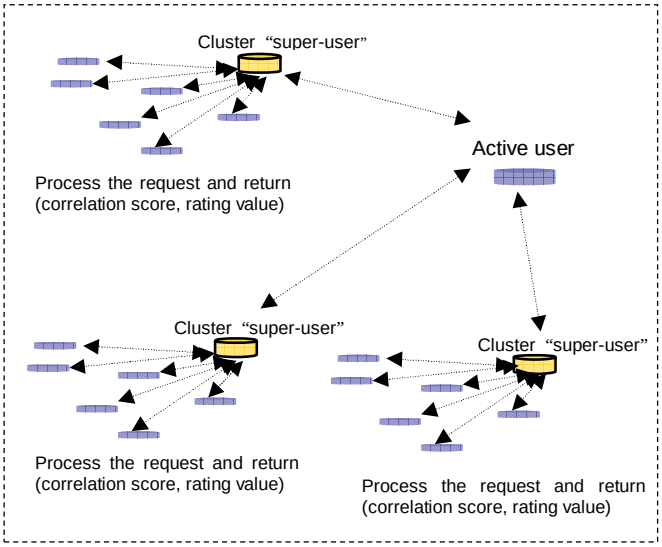

Figure 1. Hierarchical Decentralized Storage of User Profiles

Data Obfuscation Policies

The general approach of this work suggests that an active user will aggregate local recommendations generated by different super-peers into a single global recommendation. The process of generating the local recommendations within each peer-group still poses privacy concerns similar to the original one. These, however, are now on a smaller scale as local peer groups may contain only a limited number of peers. Hence, the recommendation process generates potentially less insecure communication. Nevertheless, to mitigate these concerns further, other privacy enhancing methods can be used within each peer group. One of these is the use of the data obfuscation policies proposed in [2]. In what follows, we provide a brief description of this method and motivations for using it.

Studies suggest that people are not equally protective of various elements of their private details. Hence, if users are in control of which of their private details are released, they can actively decide to release only some of them, and avoid divulging those values that they consider to be very private. Moreover, users might be willing to share modified values of certain fields of their profile because they want not to reveal their true preferences completely. Consider for instance profile information such as the list of books bought by a user. In this case the user may be concerned about revealing information about books related to their political ideas or health status. Conversely, users might readily share knowledge about books related to their scientific interests. This motivates the use of data obfuscation policies to mitigate privacy concerns. These policies change the user's profile before its similarity with the active user is calculated, effectively hiding some parts of the original user's profile. We denote three generic policies for modifying the contents of user profiles:

- *Uniform Random* obfuscation – real ratings in the user profile are substituted by random values chosen uniformly in the range of possible ratings in the dataset.
- *Bell Curved Random* obfuscation – real ratings in the user profile are substituted by random values chosen using a bell-curve distribution with properties similar to the statistical properties of the data in the dataset (e.g., average and standard deviation of the ratings).
- *Default* obfuscation(*x*) – real ratings values in the profile are substituted by a predefined constant value *x*.

For the *Default* obfuscation policy we use either values that represent extreme rating values or values that are close to the average rating of the dataset. Using extreme values in the obfuscation policy, as we shall show later, has a strong negative effect on recommendation accuracy, as it substitutes the true value, that should be close to the average, with one that is very different from the average. Moreover, this approach is very unlikely to be adopted by a user who wants to protect his privacy, since these extreme ratings will clearly show some precise polarized user preference. The *Bell Curved Random* policy reflects the actual distribution of the data and is supposed to provide the best accuracy, while preserving user privacy, since it is going to reveal a user with average preferences. Similarly, the *Uniform* policy will produce ratings that are not too far from the average behavior of the user. Hence this method is supposed to provide reasonably good privacy protection as well as recommendation accuracy. In the next section we examine the impact of these policies on the accuracy of the generated recommendations, using several known datasets having different statistical characteristics.

EXPERIMENTAL RESULTS

In order to evaluate the proposed approach, we simulated a distributed environment using a multi-threaded Java implementation, where any user can initiate a recommendation request. As described earlier, each request is transferred to a set of super-peers. The number of super-peers in the system is static, whereas the peers are seeded randomly between the super-peers. Based on a predefined parameter, each super-peer selects a random subset of the underlying peers to be queried. After receiving a request, each super-peer finds *K*-nearest neighbors by computing the

dataset	users	items	lower	upper	total rats	avRated	density	average	stddev	MAE-NP
Jester	48483	100	-10	10	3519449	72,5914	0,725914	0,816762	4,40028	0,220014
MovieLens	6040	3952	1	5	1000209	165,5975	0,041902	3,580477	0,934619	0,233655
EachMovie	74424	1649	0	1	2811718	37,77972	0,022911	0,607307	0,223402	0,223402

Table 1. Datasets properties

dataset	neutral	random	Negative	positive	average rating	variance
Jester	0	Random(-10,10)	-10	10	0,816762	4,40028
MovieLens	3	Random(1,5)	1	5	3,580477	0,934619
EachMovie	0.5	Random(0,1)	0	1	0,607307	0,223402

Table 2. Datasets properties

similarity between its underlying peers and the active user. Similarity computation (using the commonly used Mean Square Difference metric) is done locally by the peers, on possibly obfuscated profiles. Each super-peer returns to the active user an aggregated *local* rating on the relevant item jointly with the aggregated similarity of the peers in the neighborhood. Upon receiving the responses from the super-peers, the active user generates a *global* recommendation as a weighted aggregation of the super-peers' local recommendations. To measure the accuracy of the recommendation, we computed the Mean Average Error (MAE) [8] by:

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

where N denotes the total number of the generated recommendations, p_i is the i^{th} recommendation, and r_i is the real i^{th} rating. To compare the MAE values across different datasets, we normalized the MAE by dividing it by the range of possible ratings in the respective dataset.

In order to provide solid empirical evidence, we used three well-known datasets: Jester [5], MovieLens [14] and EachMovie [7]. Previous work [2] examined the issue of obfuscating user profiles on a dense matrix of users taken from the Jester dataset. In this work, we expand the experiments to include both dense and sparse datasets. Table 1 summarizes the different statistical properties of the datasets: number of users in the dataset, range of ratings, total number of ratings, average number of items rated by a user, density of the data (i.e., relative percentage of the rated items in the matrix), and statistical data about the ratings: average and standard deviation. We compared our MAE results to the MAE of a non-personalized recommendation algorithm that serves as a baseline measure. Non-personalized recommendation is computed as the average rating of the given item in the overall user population. Thus, the non-personalized MAE is computed by assigning to the predicted value p_i the constant average rating. The non-personalized MAE values were normalized by dividing them by the range of possible ratings in the respective dataset.

We performed two types of experiments. The first experiment examines the effect of data obfuscation and the second examines the effect of querying a subset of peers in the peer-group on the accuracy of the generated recommendations. For this, we used five different methods for modifying the data in user profiles, which were instantiated by the above mentioned generic obfuscation policies.

- *Positive* - substitutes the real rating with the highest positive rating in the dataset (*+10* for Jester, *5* for MovieLens and *1* for EachMovie)
- *Negative* - substitutes the real rating with lowest negative rating in the dataset (*-10* for Jester, *1* for MovieLens and *0* for EachMovie)
- *Neutral* - substitutes the real rating with neutral rating, i.e., an average between the maximal and minimal possible ratings in the dataset (*0* for Jester , *3* for *MovieLens*, and *0.5* for EachMovie)
- *Random* - substitutes the real rating with a random rating in the range of ratings in the respective dataset (between *-10* to *10* for Jester, between *1* to *5* for MovieLens and between *0* to *1* for EachMovie)
- *Distribution* - substitutes the real rating with a rating reflecting the real distribution of ratings in the dataset (in terms of average and variance).

The parameters of the above obfuscation methods are summarized in Table 2.

In the first experiment we did not employ the introduced hierarchical topology for the neighborhood formation. Thus, all the underlying peers in the group responded to a query returning an obfuscated version of their user profile. This was done to allow the general behavior of different obfuscation policies to be examined using several datasets having different statistical characteristics (as detailed in Table 1). Hence, in this experiment we measured the effect

of gradually replacing increasing elements of user profiles with either a predefined value or randomly chosen value of a given distribution. For each dataset we gradually increased the percentage of user profile that was modified (hereafter referred to as the *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 set of *10,000* users-items ratings to be recommended, and for each possible obfuscation rate we measured the MAE for the whole set. Figures *2*, *3* and *4* show MAE values as a function of the obfuscation rate. They refer to the Jester, MovieLens and the EachMovie datasets, respectively.

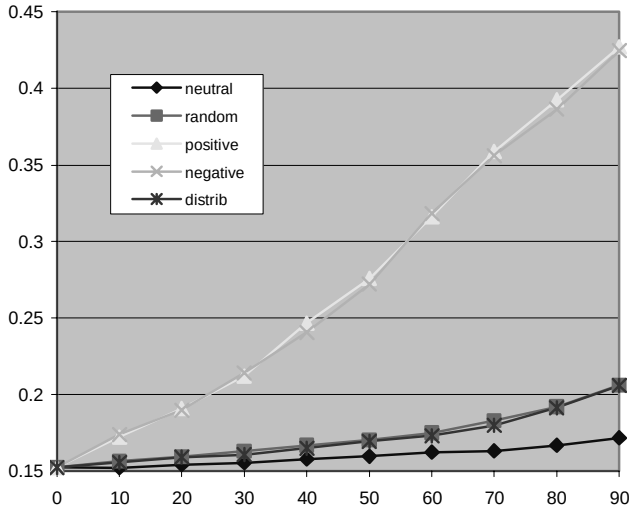

Figure 2. MAE in Jester Dataset

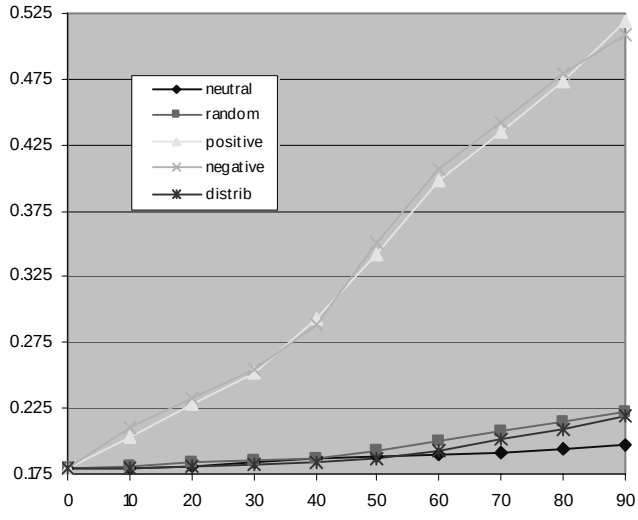

Figure 3. MAE in MovieLens Dataset

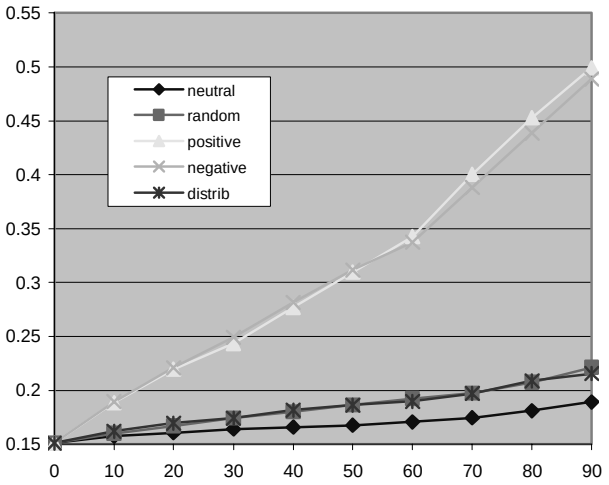

Figure 4. MAE in EachMovie Dataset

The graphs show that the effect of *Random*, *Neutral*, and *Distribution* obfuscation policies is quite similar. Obfuscating parts of the dataset according to the above policies has a minor impact on the MAE of the generated recommendations. The MAE rate slightly increases in a roughly linear manner with the obfuscation rate; however, the change is minor (in the range of *5%* to *7%*, for different datasets), and the recommendations are still accurate. This can be explained by the fact that in *Neutral*, *Distribution*, and *Random* policies, the modified values are close to the real distribution of the ratings in the dataset and the obfuscation does not significantly modify the ratings vectors of the users. Thus, substituting the actual ratings with similar ratings creates only a small overall impact on the MAE computed over many recommendations.

Conversely, in *Positive* and *Negative* obfuscation policies, the actual ratings are substituted by highly dissimilar values, as they are far from the average values in the dataset. Thus, replacing the real ratings with extremely positive or negative ratings does modify the ratings vector of the users. As a result, the generated recommendations are inaccurate and the MAE rate increases roughly linearly (in the range of *27%* to *33%*, for different datasets) with the obfuscation rate. The slope of the increase is significantly higher than in *Random*, *Neutral* and *Distribution* obfuscation policies. As can be clearly seen, this observation is true for all three datasets that were used in the experiment.

The second experiment was designed to evaluate the impact of changing the number of peers involved in the neighborhood formation within an individual peer-group (one super-peer) on the accuracy of the global recommendations. This also has implications on the scalability of our approach, as it correlates with the total number of peers involved in the recommendation generation process. Since we do not have an a-priori known topology for the super-peers, we decided to simulate a general case by arbitrarily assigning each user (peer) to a

single super-peer. When a super-peer is queried, it decides (using a predefined parameter) which subset of the underlying users should answer the query. For each dataset, we gradually increased the percentage of peers that were queried by the super-peers within the peer-groups (hereafter referred to as *keep rate*) from *10%* to *100%* of the peers. We produced a fixed set of *10,000* users-items ratings to be recommended, and for each possible obfuscation rate we measured the MAE for the whole set. To allow the behavior of choosing a subset of the peers to be examined, no obfuscation policy was used while querying the peers within a peer-group.

Figure *5* presents the MAE results as a function of the keep rate for Jester, MovieLens and the EachMovie datasets. Is illustrates that the MAE for all three datasets is close to the original value for a relatively low percentage of the queried users. For example, in Jester the MAE curve is close to the original MAE when only *30%* of the peers are queried, and in very sparse MovieLens and EachMovie datasets it occurs at approximately *40% and 50%*, respectively. Thus, it demonstrates that the number of users that should actually be queried is relatively low, and allows us to conclude that the hierarchical setting offers good scalability with the number of peers**.**

Figure 6 illustrates the decrease in the MAE compared to an MAE calculated when using 100% of the peers. It is computed from the results of Figure 5 by subtracting the calculated MAE from the MAE calculated by keeping 100% of the peers. It illustrates more clearly that the MAE converges very fast to the MAE that can be obtained using all the user profiles. It also illustrates that for Jester this convergence is quicker than for MovieLens, and the latter converges is faster than EachMovie.

Analysis

The results of these experiments show that the obfuscated recommendation results are quite similar for different datasets with different levels of density. For instance, the effect of the random policy is an increase of the MAE from the value that can be obtained with no obfuscation to a value close to the MAE of non-personalized recommendations. This seems in contrast with a common belief that the accuracy of the CF recommendations is strongly correlated to the sparseness of the dataset. Following this, we conjecture that obfuscating data in a sparse dataset would have lead to very unreliable data for recommendations generation, as the amount of initial data was already limited. Conversely, dense datasets were not expected to be impacted that strongly, as recommendations are still based on a relatively large amount of reliable data (even after performing the obfuscation).

In our experiments we used the Jester dataset, which is relatively dense (*72.59%*), and both MovieLens and EachMovie datasets, which are very sparse (density of *4.19*% and *2.29%*, respectively). Figures *2*, *3* and *4* show that the behavior of the MAE as a function of obfuscation rate for all these datasets is similar (and also similar to the dense subset of Jester used in [2]). Figures *5* and *6* show that the behavior of the MAE as a function of the number of peers chosen for all these datasets is roughly similar, where the sparse datasets needs only about *10-20%* more peers to converge to the best attainable MAE. This empirical evidence suggests that the intuitive assumption described is incorrect. We consider this to be an important observation, since the issue of not having enough ratings to create reliable recommendations (referred to in the literature as *data sparseness*) is a well-known research issue in Collaborative Filtering recommender systems [13].

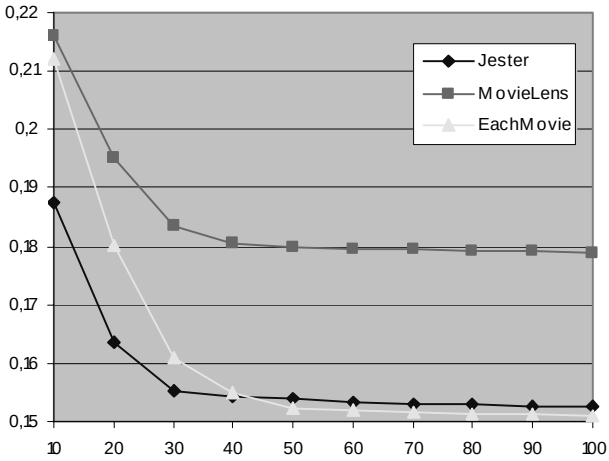

Figure 5. MAE vs. keep rate

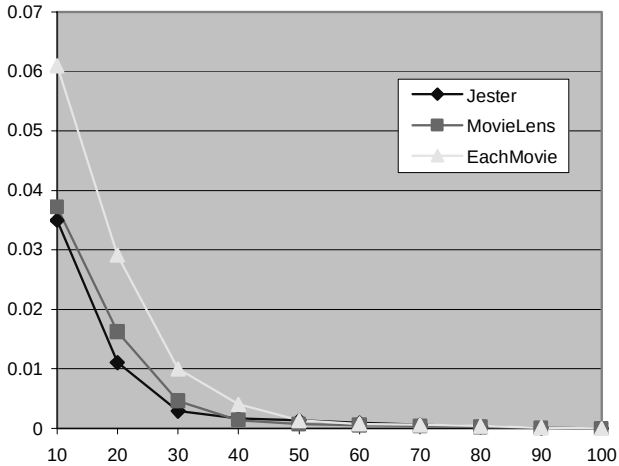

Figure 6. MAE %100 - MAE

We hypothesize that our observations could be explained by a high value of cross-users data redundancy within the datasets. This means that the users of each of the above datasets could be roughly partitioned into a small set of *classes*, such that the number of similar users in each class is relatively high. Hence, the ratings of different users within each class are highly correlated, and only a small number of representative ratings for each class is needed to

generate accurate recommendations. Data obtained by selecting additional users from a class is repetitive and therefore redundant. In this case, obfuscating even large parts of the ratings has a minor effect on the accuracy of the recommendations, since the remaining users and ratings in the class still provide a stable basis for building accurate recommendations.

Our analysis is supported by the second experiment's results, where we evaluated the effect of querying a subset of users in each peer-group in the process of recommendation generation. Our results show that it is similar for all three datasets and that a relatively small number of users is needed in order to create a reliable recommendation (*30%-40%* in our experiments). This was observed regardless of the density of the datasets and contradicted our initial assumption that the number of required users would be correlated with the density of the dataset. Hence, we believe that further work is needed to evaluate the impact of the obfuscation polices with regard to other statistical properties of the dataset (e.g., average and standard deviation of the ratings).

Threats to Validity

Care should be taken in interpreting the experimental results and the conclusions thus drawn. In what follows we describe several possible threats to the validity of our results and analysis. The first threat lies in the fact that previous work that studied people's privacy concerns dealt with broader forms of recommenders and different types of data. For example, people were either concerned about the privacy of movie ratings or they were not. However, our current evolution focuses on obfuscating data within an individual type of data (movie recommendations). Thus, it is not clear that people may be content to have a proportion of their ratings for a single class shared (e.g., movies). Thus, future work should validate that these results are applicable to broader types of data.

Secondly, our evaluation focuses mostly on generating recommendations for the average user, as we measure the error in the recommendations by using the MAE measure. However, although this is a very common measure in the CF recommender systems research domain, caution should be taken when interpreting the results which are based on this measure. This is because the data used for recommendations are usually of a "long tail" distribution (based on the power law distribution). Hence, most of the rated items are of interest to only a very small proportion of users (e.g., movies about 8th century Samurai fighting). However, due to the extreme size of the world there are still many users having such interests. Targeting recommendations for the average user will mostly not take into account the effect of losing some of these critical ratings. Hence, as there is obviously a real merit in considering such cases, they should be evaluated separately.

A third threat lies in the fact that people might like to protect their ratings in a more selective way (e.g., only some of them, and only in a particular usage context). Therefore, it would be important to understand whether these particular users' concerns are met or alleviated by the obfuscation and aggregation methods explored in this work. These issues should presumably be analyzed with the aid of social user studies, which should provide an indication of how people feel about the different obfuscation strategies and how they would want to apply them. It would be beneficial to have a quantifiable metric of whether people can understand all the policies and which of them they believe would be most useful in terms of preserving their privacy (and how). In this context, it would also be highly beneficial to measure whether people would choose to release certain particular preferences and not others. For example, they might want to hide their ratings of violent movies but would be willing to release others. In addition, there is also the question of how much accuracy loss is tolerable, since it is important for interpreting our results.

CONCLUSIONS AND FUTURE WORK

The need to protect user privacy is triggering growing research efforts. Users are concerned about their privacy and refrain from using valuable Web applications to prevent an exposure. Privacy hazards for personalization systems are aggravated by the fact that effective personalization requires large amounts of personal data. Users looking for accurate personalized information, possibly of various kinds, may well need to interact with a different set of users and systems every time. This would ensure the collection of sufficient relevant data, thereby allowing an accurate recommendation to be provided. Distributed infrastructures can be used to facilitate the development of such personalized environments

This work provides both new methodological and experimental contributions. We suggest using a notion of hierarchical topology, where the peers are organized in peer-groups managed by the super-peers. The super-peers encapsulate computations made by the underlying peers and then aggregate their results before sending them to the active user. Thus, an attacker cannot learn the properties of a single user, but rather only collect the aggregated preferences of a large group of users (managed by the given super-peer). We evaluated the approach of obfuscating user profiles using a number of publicly available datasets having different data characteristics. Our experimental results demonstrate that relatively large elements of the user profile could be obfuscated without hampering the accuracy of the generated CF recommendations. Thus, adding the proposed privacy enhancements does not severely affect the accuracy of the recommendations based on the CF algorithm.

In this work, we investigated two tangential privacy-enhancing techniques: obfuscation of the profiles and querying a subset of the available peers. Although currently we have not integrated both approaches, we see it as a natural extension for future work. We believe this may

increase the overall privacy of the CF, while keeping the generated recommendations reasonably accurate. Another research direction that we plan to take in the future is studying the effect of various topologies in peer distribution. In this work, the peers were seeded randomly among the peer-groups. This does not reflect a real-life scenario, where the peers should be clustered to the peer-groups according to various criteria, e.g., trust [10]. Using such a setting, where a query is sent only to a highly relevant subset of peer-groups, would not only optimize the communication overheads of the recommendation generation process, but would certainly reduce the privacy hazard and would probably increase user confidence in using such system.

ACKNOWLEDGMENTS

We thank the anonymous reviewers who provided insightful comments that helped us to improve our work and present it in as clear a manner as possible.

REFERENCES

1. S. Brier. “How to Keep your Privacy: Battle Lines Get Clearer.” In The New York Times, 13-Jan-97.
2. S. Berkovsky, Y. Eytani, T. Kuflik and F. Ricci. "Privacy-Enhanced Collaborative Filtering." In Workshop on Privacy-Enhanced Personalization, Edinburgh, UK, 2005.
3. J. Canny. “Collaborative Filtering with Privacy.” In IEEE Symposium on Security and Privacy, Oakland, CA, 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.L. Herlocker, J.A. Konstan, Al Borchers, and J. Riedl. “An Algorithmic Framework for Performing Collaborative Filtering.” In ACM SIGIR Conference, 1999.
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. Gilburd, A. Schuster, and R. Wolff. “k-TTP: A New Privacy Model for Large-Scale Distributed Environments.” In ACM-SIGKDD, Seattle, 2004.
10. T. Olsson. “Decentralised Social Filtering Based on Trust.” In proceedings of AAAI-98 Recommender Systems Workshop, Madison, WI, 1998.
11. H. Polat and W. Du. “Privacy-Preserving Collaborative Filtering Using Randomized Perturbation Techniques.” In proceedings of International Conference on Data Mining, Melbourne, FL, 2003.
12. H. Polat and W. Du, “Privacy-Preserving Top-N Recommendation on Horizontally Partitioned Data.” In proceedings of the WI Conference, Compiegne, 2005.
13. J. A. Konstan , B. N. Miller, D. Maltz , J., L.R. Gordon , J. Riedl. “GroupLens: Applying Collaborative Filtering to Usenet News.” In Communications of the ACM, v.40 n.3, p.77-87, 1997.
14. P. McJones. “Eachmovie Collaborative Filtering Data Set”, available online at http://research.compaq.com/SRC/eachmovie/, 1997.
15. J.B. Schafer, J.A. Konstan, J. Riedl, “E-Commerce Recommendation Applications”, Journal of Data Mining and Knowledge Discovery, vol. 5 (1/2), pp. 115-152, 2001.
16. J. Breese, D. Heckerman, and C. Kadie. “Empirical Analysis of Predictive Algorithms for Collaborative Filtering.” In Conference on Uncertainty in Artificial Intelligence, Madison, WI, 1998. Morgan Kaufmann Publisher.
17. A.F. Westin. “Freebies and Privacy: What Net Users Think”, technical report, opinion research corporation, 1999.
18. W. Nejdl, A. Löser, M Wolpers,W. Siberski,C. Schmitz, M. Schlosser, I. Brunkhorst. “Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks.” In World Wide Web Conference , Budapest, Hungary, 2003