Agent Based Middleware for Maintaining User Privacy in IPTV Recommender Services

Ahmed M. Elmisery and Dmitri Botvich

Telecommunications Software & Systems Group
Waterford Institute of Technology, Waterford, Ireland

Abstract. Recommender services that are currently used by IPTV providers help customers to find suitable content according to their preferences and increase overall content sales. Such systems provide competitive advantage over other IPTV providers and improve the overall performance of the current systems by building up an overlay that increases content availability, prioritization and distribution that is based on users' interests. Current implementations are mostly centralized recommender service (CRS) where the information about the users' profiles is stored in a single server. This type of design poses a severe privacy hazard, since the users' profiles are fully under the control of the CRS and the users have to fully trust the CRS to keep their profiles private. In this paper, we present our approach to build a private centralized recommender service (PCRS) using collaborative filtering techniques and an agent based middleware for private recommendations (AMPR). The AMPR ensures user profile privacy in the recommendation process. We introduce two obfuscation algorithms embedded in the AMPR that protect users' profile privacy as well as preserve the aggregates in the dataset in order to maximize the usability of information for accurate recommendations. Using these algorithms provides the user complete control on the privacy of his personal profile. We also provide an IPTV network scenario that uses AMPR and its evaluations.

Keywords: Privacy, Clustering, IPTV Networks, Recommender System, Multi-Agent Systems.

1 Introduction

Internet protocol television (IPTV) is one of the most fast growing services in ICT; it broadcasts multimedia content in digital format via broadband internet networks using IP packet switched network infrastructure. Differently from conventional television, IPTV allows an interactive navigation of the available items [1]. IPTV providers employ automated recommender services by collecting information about user preferences for different items to create a user profile. The preferences of a user in the past can help the recommender service to predict other items that might be interested for him in the future.

Collaborative filtering (CF) technique is utilized for recommendation purposes as one of the main tools for recommender systems. CF is based on the assumption that

R. Prasad et al. (Eds.): MOBISEC 2011, LNICST 94, pp. 64–75, 2012.

people with similar tastes prefer the same items. In order to generate recommendations, CF cluster users with the highest similarity in their interests, then dynamic recommendations are then served to them as a function of aggregate cluster interests. Thus, the more the users reveal information about their preferences, the more accurate recommendations provided to them. However at the same time the more information is revealed to the recommender service about the user profile, the lower user privacy levels can be guaranteed. This trade-off acts as a requirement when designing a recommender service using CF technique. Privacy aware users refrain from providing accurate information because of their fears of personal safety and the lack of laws that govern the use and distribution of these data. Most service providers would try their best to keep the privacy of their users. But occasionally, when they are facing bankruptcy, they might sell it to third parties in exchange of financial benefits. In the other side, many service providers might violate users' privacy for their own commercial benefits. Based on a survey results in [2, 3] the users might leave a service provider because of privacy concerns. The information collected by recommender service breaches the privacy of the users in two levels.

1. The real identity of the user is available to a central server. That server can associate the user profile which contains his private information to his real identity. This is an obvious privacy breach, considering that a user does not want to reveal the link between his real identity and his profile, yet he wants to use the service in that server.
2. If the user is not known to the server, the server can try to de-anonymize the user identity by correlating the information contained in the user profile and some information obtained from other databases [4].

In this paper we proposed an agent based middleware for private recommendation (AMPR) that bear in mind privacy issues related to the utilization of collaborative filtering technique in recommender service and allow sharing data among different users in the network. We also present two obfuscation algorithms that protect the user privacy and preserve the aggregates in the dataset to maximize the usability of information in order to get accurate recommendations. Using these algorithms, gives the user a complete control on his personal profile, so he can make sure that the data does not leaves his side until it is properly desensitized. In the rest of this paper we will generically refer to news programs, movies and video on demand contents as Items. Section 2 describes some related work. In Section 3 we introduce our private centralized recommender service scenario in IPTV network. In Section 4 we introduce the proposed obfuscation algorithms used in our framework. Section 5 describes some experiments and results based on obfuscation algorithms for IPTV network. Section 6 includes the conclusion and future work.

2 Related Work

The majority of the literature addresses the problem of privacy for recommender services based on collaborative filtering technique, Due to it is a potential source of leakage of private information shared by the users as shown in [5]. In [6] it is

proposed a theoretical framework to preserve the privacy of customers and the commercial interests of merchants. Their system is a hybrid recommender that uses secure two party protocols and public key infrastructure to achieve the desired goals. In [7, 8] it is proposed a privacy preserving approach based on peer to peer techniques using users' communities, where the community will have a 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 recommendations will be generated at client side. In [9, 10] it is suggest another method for privacy preserving on centralized recommender systems by adding uncertainty to the data by using a randomized perturbation technique while attempting to make sure that necessary statistical aggregates such as mean don't get disturbed much. Hence, the server has no knowledge about true values of individual rating profiles for each user. They demonstrate that this method does not decrease essentially the obtained accuracy of the results. Recent research work [11, 12] pointed out that these techniques don't provide levels of privacy as it was previously thought. In [12] it is pointed out that arbitrary randomization is not safe because it is easy to breach the privacy protection it offers. They proposed a random matrix based spectral filtering techniques to recover the original data from perturbed data. Their experiments revealed that in many cases random perturbation techniques preserve very little privacy. Similar limitations were detailed in [11].

3 Problem Formulation

3.1 System Model

We consider a system where PCRS is implemented as a third-party service that makes recommendations by consolidating the profiles received from multiple users. Each user has a set top box (STB) that stores his profile and host AMPR at his side. As shown in fig 1, the parties involved are the users, and the PCRS. We assume that PCRS follow the semi-honest adversary model, which is realistic assumption because the PCRS provider needs to accomplish some business goals and increase his revenues. Moreover, we assume the communication links between parties are secured by existing techniques. An IPTV provider uses this business model to reduce the required computational power, expenses or expertise to maintain an internal recommender service.

3.2 Design Goals

There are two requirements should be satisfied in the previous system model:

- IPTV providers care about the privacy of their catalogue which is considered an asset for their business. In the meantime they are willing to offer real users' ratings for different masked items to offer better recommendations for their users and increase their revenues.
- In the other side, privacy aware users worry about the privacy of their profiles, as sending their real ratings harm their privacy.

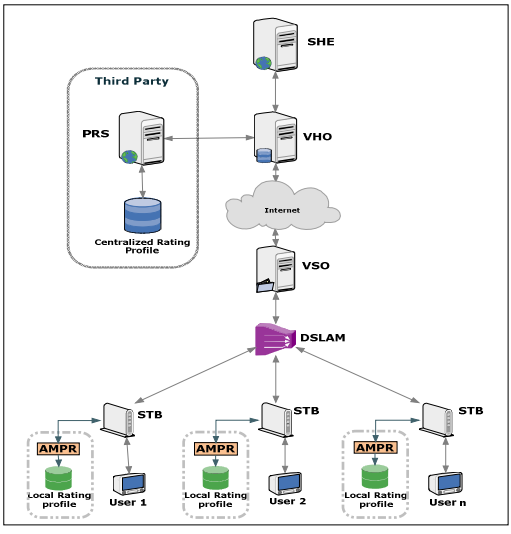

Fig. 1. Illustration of proposed combined IPTV Network

The AMPR employs two obfuscation algorithms that provide the users the required privacy level before submitting the profiles to the PCRS. Note that, we alleviate the user identity problems by using anonymous pseudonyms identities for users.

3.3 Threat Model

In this paper, AMPR provides a defence mechanism against the threat model proposed in [13] where the attacker colludes with some users inside the network to obtain some partial information about the process used to obfuscate the data and/or some of the original data items themselves. The attacker can then use this partial information for the reverse engineering of the entire data set.

4 Solution

In the next sections, we will present our proposed framework for preseving the privacy of customers' profiles show in fig 2.

4.1 PCRS Components

As show in fig 2, PCRS maintains a set data stores. The first data store is the masked catalogue of items that have been hashed using IPTV provider key or a group key. The second data store is the obfuscated users' profiles which contain users' pseudonyms and their obfuscated ratings and finally a peer cache which is an updated database about peers participated in previous recommendations formulation. The peer cache is updated from peer list database at client side. The PCRS communicates with the user through a manager unit. Finally, the clustering manager is the entity responsible for building recommendations model based upon the obfuscated ratings database.

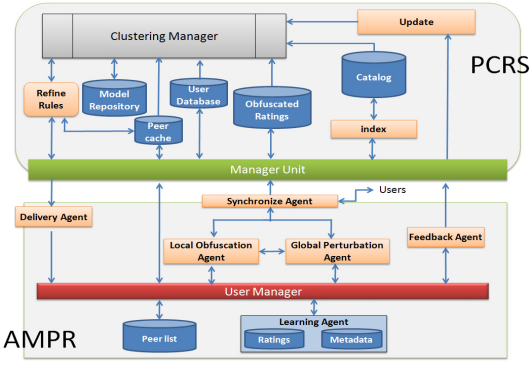

Fig. 2. PCRS framework

4.2 AMPR Components

The AMPR in the user side consists of different co-operative agents. Learning agent captures user preferences about items explicitly or implicitly to build a rating table and meta-data table. The local obfuscation agent implements CBT obfuscation algorithm to achieve user privacy while sharing the data with other users or the system. The global perturbation agent executes G-algorithm on the locally obfuscated collected profiles. These algorithms act as wrappers that obfuscate items' ratings before they are fed into the PCRS. Since the database is dynamic in nature, the local obfuscation agent desensitizes the updated data periodically, then synchronize agent send it to other users and PCRS. So the recommendations are made on the most recent ratings. More details about the recommendation process described in the next sub-section.

4.3 The Recommendation Process

The recommendation process based on the two stage obfuscation algorithms can be summarized as following more details can be found in [14]. The target user broadcasts message to other users in the IPTV network to request starting the recommendations process or update their centralized rating profiles stored at PCRS. The individual users who decided to participate in that process use the local obfuscation agent to perform *CBT* algorithm of their local rating profiles. They agree on same parameters, and then they submit their locally obfuscated profiles to the requester. The target user instructs his obfuscation agent to start *G* algorithm on the collected locally obfuscated profiles. After finishing the previous step, the target user submits all profiles to PCRS in order to receive recommendations.

5 Proposed Algorithms

In the next sub-sections, we provide two different algorithms that used by our agents to obfuscate the user profile in a way that secure his ratings in the un-trusted PCRS with minimum loss of accuracy. In our framework, each user has two datasets representing his/her profile. First one is the local rating profile which is perturbed

before merging it with similar users' profiles that rare willing to collaborate with him as part of the recommendation process. The second one is the centralized rating profile which is the output of the two obfuscation algorithms where the user can get recommendation directly from the PCRS based on it. We perform experiments on real datasets to illustrate the applicability of our algorithms and the privacy and accuracy levels achieved using them.

5.1 Local Obfuscation Using CBT Algorithm

We propose a new obfuscation algorithm called clustering based transformation (CBT) that have been designed especially for the sparse data problem in user profile. It is inspired from the block cipher idea in [15]. We present a new technique to build a transformation lookup table (TLUT) using clustering technique then approximate each point in the data set to the nearest representative value in the TLUT (the core-point for the cluster it belong to) with the help of similarity measures. The output of our obfuscation algorithm should satisfy two requirements:

1. Reconstructing the original data from the obfuscated data should be difficult, in order to preserve privacy.
2. Preserve the similarity between data to achieve accurate results.

We use local learning analysis (*LLA*) clustering method proposed in [16] to create the *TLUT*. It is important to attain an optimized *TLUT* because the quality of the *TLUT* obviously affects the performance of the transformation. *LLA* builds an initial *TLUT* and repeats the iteration till two conditions satisfied:

1. The distance function $d(x, c_i)$ between a point x and its corresponding value (core-point) c_i is minimized.
2. The distortion function between each dataset and its nearest value (core-point) becomes smaller than a given threshold.

CBT algorithm consists of following steps:

1. The user ratings stored as dataset D of c rows, where each row is sequence of fields $\mathrm{X} = \mathrm{x}_1\ \mathrm{x}_2\ \mathrm{x}_3 \ldots\ldots \mathrm{x}_m$.
2. User ratings dataset D is portioned into $\mathrm{D}_1\ \mathrm{D}_2\ \mathrm{D}_3 \ldots\ldots \mathrm{D}_n$ datasets of length L, if total number of attributes in original is not perfectly divisible by L then extra attributes is added with zero value which does not affect the result and later it is removed at step 5.
3. Generate *TLUT* using *LLA* algorithm, *LLA* takes Gaussian Influence function as the similarity measure. Influence function between two data points x_i and x_j is given as

$$f_{Gauss}^{x_i}(x_j) = e^{-\frac{d(x_i, x_j)^2}{2\sigma^2}} \tag{1}$$

While the field function for a candidate core-point given by:

$$f_{Gauss}^{D}(x_j) = \sum_{s=1}^{k} e^{-\frac{d(x_j, x_{is})^2}{2\sigma^2}} \tag{2}$$

Clustering is performed on each dataset D_i, resulting to k clusters $C_{i1}, C_{i2}, C_{i3}, ..., C_{ik}$ and each cluster is represented by its core-points, i.e. core-point of j^{th} cluster of i^{th} dataset is $(C_{ij}) = \{c_1, c_2, c_3, ...c_L\}$. Every single row portion falls in exactly one cluster. And The TLUT = (core-point (C_{i1}), core-point(C_{i2}), core-point(C_{i3}) ..,core-point (C_{ik}))

4. Each dataset D_i is transformed into new dataset D_i' using generated TLUT, each portion $Y_i = x_{(i-1)L+1}\ x_{(i-1)L+2}\ x_{(i-1)L+3} \ldots\ldots x_{iL}$ replaced by the nearest cluster core-point Z_i = core-point (C_{ij}) in which it falls.

$$Y_i \xrightarrow{transofrmed} Z_i$$

5. The transformation function is: $T(Y_i) = \{core - point(C_j \leftrightarrow d(Y_i, core - point(C_j) < d(Y_i, core - point(C_Z)\ \forall\ Z\}$
6. Now all the n transformed portions of each point are joined in the same sequence as portioned in step 2 to form a new k dimension transformed row data which replaces the X in the original dataset. In this way perturbed dataset D_i' is formed from original dataset D
7. Compute the privacy level by calculating the difference between the original dataset and transformed dataset using Euclidean distance:

$$\Pr ivacy - Level = \frac{1}{mn}\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left|x_{ij} - y_{ij}\right|^2} \tag{3}$$

5.2 Global Perturbation Using *G* Algorithm

After executing the local obfuscation process, the global perturbation algorithm at the requester side is started. The idea is cluster multidimensional data using fast density clustering algorithm, then perturb each dimension in each cluster in such a way to preserve its range. In order to allow the global perturbation agent to execute *G* algorithm, we introduce an enhanced mean shift (EMS) algorithm which is tailored algorithm for the global perturbation phase that has advantage over previous algorithm proposed in [17] and it requires low computational complexity in clustering large data sets. we employ Gaussian *KD-tree* [18] clustering to reduce the feature space of the locally obfuscated data.

The G algorithm consists of two steps:

Step 1: Build different density based clusters

1. We build the tree in a top down manner starting from a root cell similar to [18, 19]. Each inner node of the tree S represents a *d*-dimensional cube cell which stores the dimension S_d along which it cuts, the cut value S_{cut} on that dimension, the bounds

of the node in that dimension S_{min} and S_{max}, and pointers to its children S_{left} and S_{right}. All points in the leaf nodes of the *kd* tree are then considered as a sample and the *kd-tree* stores *m* samples defined as $y_j^*, j = 1, \ldots, m$ that construct the reduced feature space of the original obfuscated data set.

2. Assign each record x_i to its nearest y_j based on *kd*-search, then compute a new sample, we called it $y_j^*, j = 1, \ldots, m$.
3. Generated y_j^* is a feature vector of *d*-dimensions, that is considered as a more accurate sample of the original obfuscated data set that will be used in the mean shift clustering.
4. The mean shift clustering iteratively performs these two steps:
   - Computation of mean shit vector based on the reduced feature space as following:

$$m(x_j) = \frac{\sum_{y_i^* \in N(y_j^*)} y_i^* g\left(\left\|\frac{x_j - y_i^*}{h}\right\|^2\right)}{\sum_{y_i^* \in N(y_j^*)} g\left(\left\|\frac{x_j - y_i^*}{h}\right\|^2\right)} - x_j, j = 1, 2..m \tag{4}$$

   Where $g(x) = -k'(x)$ defined when the derivate of function $k(x)$ exists, and $k(x), 0 \le x \le 1$ is called kernel function satisfying: $k(x) = c_{k,d} k\left(\|x\|^2\right) > 0$, $\|x\| \le 1$ and $\int_{-\infty}^{\infty} k(x) dx = 1$

   - Update the current position x_{j+1} as following:

$$m(x_{j+1}) = \frac{\sum_{y_i^* \in N(y_j^*)} y_i^* g\left(\left\|\frac{x_j - y_i^*}{h}\right\|^2\right)}{\sum_{y_i^* \in N(y_j^*)} g\left(\left\|\frac{x_j - y_i^*}{h}\right\|^2\right)}, j = 1, 2..m \tag{5}$$

   Until reaching the stationary point which is the candidate cluster centre. x_j will coverage to the mode in reduced feature space, finally we get approximate modes of original data defined as $z_x, x = 1, ..., k$.
5. Finally, the points which are in the mode are associated with the same cluster. Then we interpolate the computed modes in samples to the original obfuscated data by searching for the nearest mode z_x for each point x_i .

Step 2: Generating random points in each dimension range

For each cluster C , perform the following procedure.

1. Calculate the interquartile range for each dimension A_i .
2. For each element $e_{ij} \in A$, generate a uniform distributed random number r_{ij} in that range and replace e_{ij} with r_{ij} .

6 Experiments

The proposed algorithms are implemented in C++. We used message passing interface (MPI) for a distributed memory implementation of *G* algorithm to mimic a reliable distributed network of peers. We evaluated the proposed algorithms from two different aspects: privacy achieved and accuracy of results. The experiments presented here were conducted using the Movielens dataset [20]. The dataset contains users' ratings on movies using discrete values between 1 and 5. We follow the experiential scenarios presented in [14] We divide the data set into a training set and testing set. The training set is obfuscated then used as a database for the PCRS. Each rating record in the testing set is divided into a rated items t_u and unrated items r_u. The set $t_{u,i}$ is presented to the PCRS for making predication $p_{u,i}$ for the unrated items $r_{u,i}$ using the same algorithm in [21]. To evaluate the accuracy of generated predications, we used the mean average error (MAE) metric proposed in [22]. The first experiment performed on CBT algorithm to measure the impact of the varying portion size and number of core-points on privacy levels of the transformed ratings. To measure that we kept portion size constant with different number of core-points and then we vary portion size with constant number of core-points. Based on the results shown in figs (3) and (4), we can conclude that the privacy level increases when portion size is increasing. On the other hand, privacy level is reduced with increasing number of core-points as large number of rows used in *TLUT*. Each user in the network can control his privacy by diverging different parameters of *LLA* algorithm. Note that reducing the privacy level means less information loss in the collected ratings presented to PCRS. However this means the transformed ratings are similar to the original ratings, so the attacker can acquire more sensitive information.

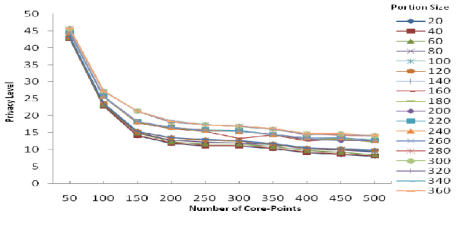

Fig. 3. Privacy level for different no.of core point

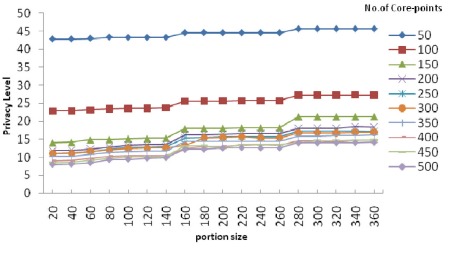

Fig. 4. Privacy level for different portion size

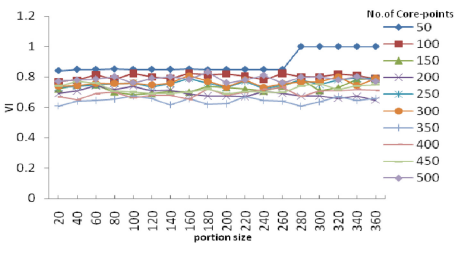

Fig. 5. VI for different portion size

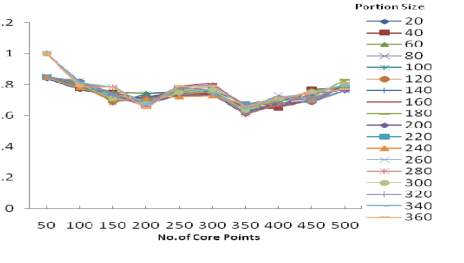

Fig. 6. VI for different no.of core points

To measure the distortion of the results, we use variation of information (VI) metric. Fig (5) shows VI for different number of core-points. One can see that at lower values of number of core-points the VI is high but slowly decreases with the increase in the number of core-points. At a certain point it rises to a local maxima then it decreases. Finally it rises again with the increasing number of core-points. We can justify that VI is high with fewer number of core-points as any point can move from one core-point to another. Moreover, with a plenty number of core-points there is a little chance of a point to move from one core-point to another, which causes increasing in VI values. The second experiment is performed on *G* algorithm to measure the impact of sample size on the accuracy level. We set a specific threshold value (100 users) for the minimum number of responding users for recommendation request. Otherwise the target user uses the PCRS obfuscated ratings database. As shown in fig (7) the increase in sample size leads to higher accuracy in the predications. However at a certain sample size, the accuracy of the predications starts decrement again due to the data loss in the sampling process.

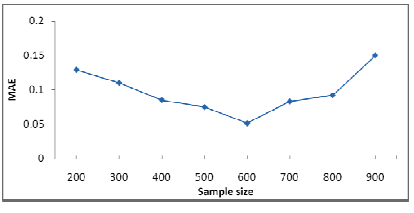

Fig. 7. Relation between sample size and MAE

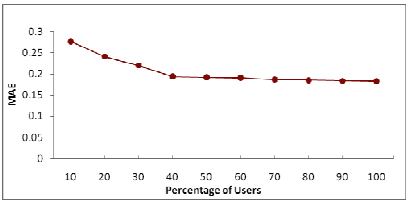

Fig. 8. Relation between Users and MAE

In the third experiment, we want to measure the impact of changing number of users involved in the group formation on the accuracy of the recommendations. We simulate a general case where the number of users was fixed to be 50.000. Then we assign different number of users to a certain recommendation request, and gradually increased the percentage of users who joined the request from 10% to 100% of them. We fixed the parameters for CBT and G algorithms then we measure MAE for the results. As shown in fig (8), the MAE value occurs at approximately 40% of the users are close to the MAE value for all users. Our conclusion is that, for low percentage of users the MAE value is close to the original MAE value for all users. As a result the target user does not need to broadcast the request to the full IPTV network to attain accurate results but he can employ multicast for certain users stored in his peer list to reduce the load in the network traffic. To illustrate the decrement of MAE values for recommendations based on diverse percentages of users groups and the whole users in the network, we calculated and plot fig (9). This verifies our conclusion that MAE approximately converges to the MAE which obtained using the whole users in our case. The final experiment was conducted to measure the impact of using CBT algorithm as a pre-processing step for G algorithm. As presented in fig (10), using CBT increases MAE values for lower percentages of participated users compared to using G algorithm alone. This can be explained, as the distortion effect of CBT algorithm will be clearly visible for a lower percentage of participating users.

However with the augment of percentage of users scale down the error in MAE values. So we can say that using the two stage obfuscation algorithms can increase the recommendation accuracy compared to only one stage based on one algorithm only.

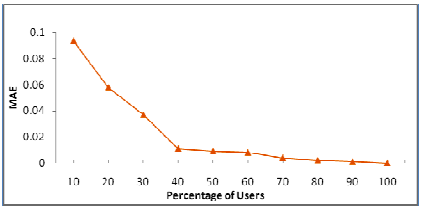

Fig. 9. The decrement of MAE values

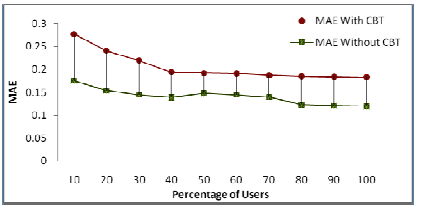

Fig. 10. The Influence of applying CBT algorithm

The results presented in these experiments show that the resulting dataset from our two stage obfuscation processes are quite similar in the accuracy of the generated recommendation to the original dataset. Our results also clarify that the proposed algorithms preserve the utility of the data to some degree such that to create reliable recommendations the target user does not have to collect profiles from numerous users, only a small percentage from users is need to attain that goal.

7 Conclusion and Future Wok

In this paper, we presented our ongoing work on building an agent based middleware to achieve privacy in recommender services. We gave an overview over the system components and recommendations process. Also we presented the novel algorithms that provide to users complete privacy over his profile privacy using two stage obfuscation processes. We test the performance of the proposed algorithms on real dataset and report the overall accuracy of the recommendations based on different privacy levels. The experiential results shows that preserving users' data privacy for in collaborative filtering recommendation system is possible and the mean average error can be reduced with proper tuning for algorithms parameters and large number of users. We need to perform extensive experiments in other real data set from UCI repository and compare the performance with other techniques, also we need to consider different data partitioning techniques, identify potential threats and add some protocols to ensure the privacy of the data against these threats.

References

1. Hand, S., Varan, D.: Interactive Narratives: Exploring the Links between Empathy, Interactivity and Structure, pp. 11–19 (2008)
2. Cranor, L.F.: 'I didn't buy it for myself' privacy and ecommerce personalization. In: Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society. ACM, Washington, DC (2003)
3. Dialogue, C.: Cyber Dialogue Survey Data Reveals Lost Revenue for Retailers Due to Widespread Consumer Privacy Concerns. Cyber Dialogue (2001)

4. Narayanan, A., Shmatikov, V.: Robust De-anonymization of Large Sparse Datasets. In: Proceedings of the 2008 IEEE Symposium on Security and Privacy. IEEE Computer Society (2008)
5. McSherry, F., Mironov, I.: Differentially private recommender systems: building privacy into the net. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 627–636. ACM, Paris (2009)
6. Esma, A.: Experimental Demonstration of a Hybrid Privacy-Preserving Recommender System. In: Gilles, B., Jose, M.F., Flavien Serge Mani, O., Zbigniew, R. (eds.), 161–170 (2008)
7. Canny, J.: Collaborative filtering with privacy via factor analysis. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 238–245. ACM, Tampere (2002)
8. Canny, J.: Collaborative Filtering with Privacy. In: Proceedings of the 2002 IEEE Symposium on Security and Privacy, p. 45. IEEE Computer Society (2002)
9. Polat, H., Du, W.: Privacy-Preserving Collaborative Filtering Using Randomized Perturbation Techniques. In: Proceedings of the Third IEEE International Conference on Data Mining, p. 625. IEEE Computer Society (2003)
10. Polat, H., Du, W.: SVD-based collaborative filtering with privacy. In: Proceedings of the 2005 ACM Symposium on Applied Computing, pp. 791–795. ACM, Santa Fe (2005)
11. Huang, Z., Du, W., Chen, B.: Deriving private information from randomized data. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 37–48. ACM, Baltimore (2005)
12. Kargupta, H., Datta, S., Wang, Q., Sivakumar, K.: On the Privacy Preserving Properties of Random Data Perturbation Techniques. In: Proceedings of the Third IEEE International Conference on Data Mining, p. 99. IEEE Computer Society (2003)
13. Parameswaran, R., Blough, D.M.: Privacy preserving data obfuscation for inherently clustered data. Int. J. Inf. Comput. Secur. 2, 4–26 (2008)
14. Elmisery, A., Botvich, D.: Private Recommendation Service For IPTV System. In: 12th IFIP/IEEE International Symposium on Integrated Network Management. IEEE, Dublin (2011)
15. Blaze, M., Schneier, B.: The MacGuffin block cipher algorithm, pp. 97–110 (1995)
16. Elmisery, A.M., Huaiguo, F.: Privacy Preserving Distributed Learning Clustering Of HealthCare Data Using Cryptography Protocols. In: 34th IEEE Annual International Computer Software and Applications Workshops, Seoul, South Korea (2010)
17. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on Information Theory 21 (2003)
18. Xu, K., Li, Y., Ju, T., Hu, S.-M., Liu, T.-Q.: Efficient affinity-based edit propagation using K-D tree. In: ACM SIGGRAPH Asia 2009 Papers, pp. 1–6. ACM, Yokohama (2009)
19. Adams, A., Gelfand, N., Dolson, J., Levoy, M.: Gaussian KD-trees for fast high-dimensional filtering. ACM Trans. Graph. 28, 1–12 (2009)
20. Lam, S., Herlocker, J.: MovieLens Data Sets. Department of Computer Science and Engineering at the University of Minnesota (2006)
21. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 230–237. ACM, Berkeley (1999)
22. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22, 5–53 (2004)