Privacy Preserving Collaborative Filtering using Data Obfuscation

Rupa Parameswaran
Georgia Institute of Technology
School of Electrical and Computer Engineering
Atlanta, GA
rupa@ece.gatech.edu

Douglas M Blough
Georgia Institute of Technology
School of Electrical and Computer Engineering
Atlanta, GA
doug.blough@ece.gatech.edu

Abstract

Collaborative filtering (CF) systems are being widely used in E-commerce applications to provide recommendations to users regarding products that might be of interest to them. The prediction accuracy of these systems is dependent on the size and accuracy of the data provided by users. However, the lack of sufficient guidelines governing the use and distribution of user data raises concerns over individual privacy. Users often provide the minimal information that is required for accessing these E-commerce services. In this paper, we propose a framework for obfuscating sensitive information in such a way that it protects individual privacy and also preserves the information content required for collaborative filtering. An experimental evaluation of the performance of different CF systems on the obfuscated data proves that the proposed technique for privacy preservation does not impact the accuracy of the predictions.

The proposed framework also makes it possible for multiple E-commerce sites to share data in a privacy preserving manner. Problems such as the cold-start scenario faced by new E-commerce vendors, and biased results due to insufficient users, are resolved by using a shared CF server. We describe a centralized CF server model in which a centralized CF server makes recommendations by consolidating the information received from multiple sources.

1. Introduction

In the presence of information overload, scanning through all the available choices can be cumbersome. Humans make most decisions based on recommendations from a set of peers or seek out help from a professional. Collaborative Filtering (CF) systems automate the recommendation process by seeking out similar users and using the preferences of the common set of users to make recommendations regarding articles or items of potential interest to them [24]. Early CF systems required users to seek information from a known set of users. Automated CF systems (ACF) arose with the development of information retrieval techniques. These systems provide the user with recommendation without the user having to seek information [9].

Other developments in CF systems involved the improvement from a completely memory-based approach using nearest neighbor techniques to a model-based approach using methods like Bayesian clustering. Although several CF systems have been implemented, the improvements in the accuracy of predictions have only been marginal. In order to provide personalized information to a user, the CF system needs to be provided with sufficient information regarding his or her preferences, behavioral characteristics, as well as demographic information of the individual. The accuracy of the recommendations is dependent largely on how much of this information is known to the CF system. However, this information can prove to be extremely dangerous if it falls in the wrong hands.

The concerns over personal privacy create a limitation on the amount of information that can be provided to a CF system. Individuals refrain from providing information because of fears of personal safety. The lack of laws governing the use and distribution of this data is one of the prime reasons for these concerns. The accuracy of CF systems is limited by sparse data. The results of a survey on personal privacy [5] indicate that more than 81% of the people in the survey were willing to provide information as long as their privacy was guaranteed. The implementation of a privacy preserving framework for protecting user information is a step in this direction.

In the past decade, data obfuscation techniques have been proposed for privacy preserving mining of data. Data obfuscation techniques desensitize the original data by transformations such as the addition of random noise [1], partial suppression [25], swapping [21], and linear transformation [15][16]. In all of these approaches, the resulting data is different from the original data and cannot be mapped to its original form. Data obfuscation techniques perform the transformation in such a way that the aggre-

0-7695-3032-X/07 $25.00 © 2007 IEEE
DOI 10.1109/GrC.2007.133

gates are still preserved in the dataset. In this paper, we evaluate the feasibility of applying different data obfuscation techniques to CF and study their impact on the prediction accuracy. Since most CF systems use a similarity measure for predicting user preferences, we propose the use of a *Nearest Neighbor Data Substitution* (NeNDS) approach [18] to CF systems for protecting the privacy of user data. We also propose a privacy preserving framework for CF that allows sharing of data among multiple sellers.

The privacy preserving framework proposed here maximizes the usability of information provided by the users without violating their privacy. The User-information database as well as Ratings information database are obfuscated in such a way that clusters of similar data are preserved while hiding the actual values of the data. The obfuscated data are sent to a centralized CF server for making predictions, which are then sent back to the corresponding E-commerce vendors. The obfuscated results are used to make recommendations to the users. This is the first approach that provides a robust privacy protection framework that allows information regarding user demographics and ratings to be shared among multiple vendors.

2. Related Work

The term 'Collaborative Filtering' (CF) was first introduced in the Tapestry system [6], for filtering electronic documents through e-mail and Usenet postings. In this system, a user explicitly requests recommendations based on reviews of a specific set of known individuals. The drawback of this system is that it requires a close-knit group of people who are aware of each other's interests. The lack of scalability of this system for larger networks led to the development of more Automated Collaborative Filtering systems (ACF) [23]. The GroupLens CF system [22] pioneered the research on ACF by using pseudonymous users to provide ratings for movies and Usenet news articles. Some of the other recommendation systems such as the e-mail based music recommendation system [27], Ringo, and the web-based movie recommendation [10], Video Recommender, also developed ACF algorithms for recommendations. All three systems use neighborhood-based prediction algorithms such as Pearson's correlation and vector similarity. These algorithms are referred to as memory-based algorithms because they use the raw data in the database to make recommendations. Model-based approaches such as Bayesian network models and cluster-based models were proposed in [26][4]. These algorithms first develop cluster-based models or Bayesian network models on the database. The models are then used for making predictions for users on items that have not yet been rated by them. This makes model-based CF algorithms faster and less memory-intensive. Hybrid memory-model based approaches have also been developed to improve accuracy of predictions [19].

As with any system that stores personal information of individuals, CF systems are vulnerable to privacy invasion. Although meta-store fronts such as *Amazon*, *C-net*, *Yahoo* assert privacy policies that protect user data, their policies are intentionally vague in certain areas. For instance, Amazon's policy states that in the event that the company is bought over, the personal assets are subject to be transferred to the parent company. Such loop holes in the policies present privacy concerns resulting in users refraining from divulging any personally identifiable information. This results in incomplete or sparse databases. The absence of complete information or dense databases affects the accuracy of the recommendation systems. Privacy preservation by factor analysis [2][3] proposes a secure computation technique using homomorphic encryptions. Here users' ratings are stored as encrypted vectors and aggregates of the data are provided in the public domain. This approach requires the users to seek out recommendations explicitly. The random perturbation approach proposed in [20] uses a noise vector to mask the original data. Although the technique permits heterogeneous diffusion based recommendations, the accuracy of the predictions is dependent on the amount of noise added. The drawbacks of random perturbations are discussed in [18]. In this paper, we propose a frame-work for shared privacy preserving collaborative filtering using a hybrid NeNDS based data obfuscation approach.

Secure recommendations using trust-based CF techniques have been proposed in [13][14] to protect against targeted attacks to push a chosen set of items. Such attacks, known as shilling attacks [28][12] are achieved by introducing false profiles in the database that rate a chosen set of items in such a way that their overall rating changes significantly. Trust-based systems prevent such attacks by introducing a web of trusted users whose ratings are preferred over the un-trusted users. While trust-based systems protect the truthfulness of the ratings and avoid attacks on the CF system, the privacy framework attempts to protect the personally identifiable fields of individuals participating in the ratings. A secure CF system should protect the quality of the recommendations as well as the privacy of the participants that provide the ratings.

3. The Privacy Framework

The model for privacy preserving collaborative filtering is explained in detail in this section. The privacy framework serves as a wrapper that obfuscates the relevant fields of data before they are fed to the CF engine. A diagrammatic view of the model is shown in Figure 1 using an example having three meta-store fronts $[MS_1, MS_2, MS_3]$ such as

Amazon, *C-net*, *Yahoo* that wish to share information in a privacy preserving way. Each MS_i's has three databases, a *User-info* database that stores demographic information regarding its users, an *Item-info* database that stores information regarding the items in its inventory, and a *Ratings-info* database that stores information regarding the ratings provided by the users on the items purchased. The databases are obfuscated and sent to the central CF server. The CF engine combines the information from all three meta-store fronts and creates three aggregated databases as shown. Recommendations are made for all the unrated items for each record in the ratings database. The aggregate database is then divided back into the three individual databases, which are now populated with recommendations for unrated items. The databases are then sent back to the meta-store fronts. The stores provide recommendations to their users based on the results obtained from the CF engine. Since the databases are dynamic in nature, the MS_i obfuscate the updated databases periodically and send them to the CF server so that the recommendations are made on the most recent ratings of individuals. This type of framework allows different e-commerce vendors to share proprietary information about their customers without violating their privacy. Providing a secure framework for shared collaborative filtering is one of the contributions of this paper.

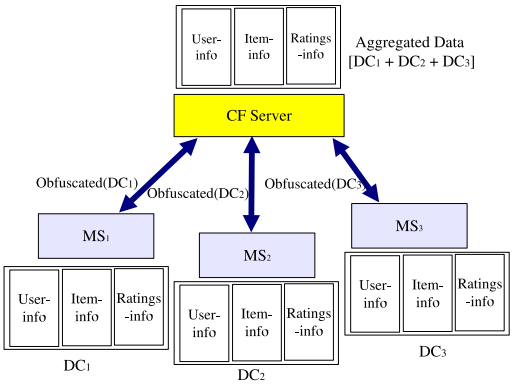

Figure 1. Privacy preserving framework for CF.

3.1. Data Obfuscation

Several approaches have been proposed for privacy preserving data mining applications. Random data perturbation [1] and data anonymization [25] are some commonly used data obfuscation techniques for applications where aggregate statistics are sufficient. These approaches protect data by adding random noise to them or by a process of suppression and generalization. The lossy nature of the transformations destroys the inherent clustering in the data, making them unsuitable for applications that use classification or cluster-based data mining. Geometric transformations [15][16] and data swapping [21] preserve clustering, but offer weak privacy preservation of the data, which renders them unsuitable for sensitive applications [18]. In NeNDS, each field of the database is treated separately, and the datasets are obfuscated by permuting sets of similar items. The permutation process ensures lossless transformation and also offers a stronger transformation than data swapping. Permutation among similar elements ensures that the clusters are preserved. A comparison of the strength of the data obfuscation techniques with respect to privacy protection and data usability is presented in [18]. The results show that NeNDS offers robust data privacy as well as data usability. The approach can be applied on any dataset that forms a metric space. One drawback of NeNDS is that the transformed data might be close enough to the original value to be considered vulnerable. This vulnerability is fixed by performing a geometric transformation such as rotation, scaling, or translation on the NeNDS-obfuscated data. The linearity property of geometric transformations preserves clustering and changes the values of the individual data. The weakness of geometric transformations is taken care of by performing NeNDS-based data obfuscation as a first step. This hybrid-NeNDS approach is used here to obfuscate the data for CF.

Table 1 represents a *Ratings-info* database (DB) with ratings of the 8 users on 2 items. The ratings are on a scale of $[1 - 10]$. These ratings are first transformed using NeNDS by creating 2 neighborhoods for each dataset and then permuting the data in each neighborhood. The NeNDS transformed data are presented in Table 2. The missing entries in the database are not included in any neighborhood and are retained even after transformation. The NeNDS-transformed database is then scaled by a factor of 0.8 on both fields. The transformed database is shown in Table 3.

The accuracy of the predictions for large databases is studied in Section 4. For shared CF, each meta-store front performs a NeNDS transformation on the data followed by a geometric transformation using the same parameters. The parameters for the geometric transformations can be decided by the central server, or by a secure token exchange among the meta-store fronts. Rotation-based transformations cannot be used here because of the presence of incomplete records. Rotation of a record with a missing entry results in a transformed record that has a non-zero value in place of the missing entry. Rotation of such records distort the relative distances between records. CF systems using similarity measures for predicting user preference would fail if the relative distances are altered significantly. Since most of the databases used for CF are sparse databases, rotation-based transformations are not feasible. The scal-

ID	Rating-1	Rating-2
1	4	3.5
2	5.5	4.1
3		2.5
4	9	7.5
5	8.5	8
6	4.5	
7	9.5	9
8	10	9.5

Table 1. Original DB

ID	Rating-1	Rating-2
1	4.5	2.5
2	4	3.5
3		4.1
4	8.5	9
5	10	7.5
6	5.5	
7	9.5	9.5
8	9	8

Table 2. NeNDS-tranformed DB

ID	Rating-1	Rating-2
1	3.6	2
2	3.2	2.8
3		3.2
4	6.8	7.2
5	6.4	6
6	3.8	
7	7.6	7.6
8	7.1	6.4

Table 3. Scaled-NeNDS trasformed DB

ing transformation discussed here can be replaced by any linear transformation vector that is not affected by missing entries.

3.2. Privacy Analysis of NeNDS

The data privacy provided by NeNDS and GT-NeNDS are analysed in this section. The privacy provided by a data obfuscation technique is measured in terms of its *reversibility* property [18]. *Reversibility* is dependent on the minimum number of records r that are sufficient for complete reverse engineering.

In the case of *NeNDS*, complete reversal of the entire data set would require the knowledge of at least $r = c - 1$ distinct dat a elements for each neighborhood, where c is the minimum size of a neighborhood. Even partial reversal of a single neighborhood w ould require the knowledge of $c - 1$ of its elements. The fraction $\frac{c_i-1}{c_i}$ determines the ease of reversal of a specific neighborhood i having exactly c_i elements. The proof for this claim is provided below. The goal of the attacker is to retrieve the original value corresponding to one of the obfuscated items in a dataset with absolute certainty. We refer to this as a targe ted value attack.

Theorem 1. *Let $[X, Y]$ be the original and obfuscated datasets of size n respectively.*

$$X = x_1, x_2, \ldots, x_n \tag{1}$$

$$Y = y_1, y_2, \ldots, y_n \tag{2}$$

Let $y_t | y_t \in Y$ be the obfuscated item whose original value x_t the attacker wants to retrieve and let x_t belong to the p^{th} neighborhood. Assume that all c items in the p^{th} neighborhood are distinct values. Assume that the attacker has com plete knowledge of the NeNDS algorithm, including the value of neighborhood size c used to produce Y, but no additional knowled ge except for a subset of the original data items. Then, the attacker needs to know at least $c - 1$ original data items other than t he targeted item to succeed in a targeted value attack.

Proof. Let $[X_p, Y_p]$ be the original and obfuscated data items in the p^{th} neighborhood.

$$X_p = x_{p1}, x_{p2}, \ldots, x_{pc} \tag{3}$$

$$Y_p = y_{p1}, y_{p2}, \ldots, y_{pc} \tag{4}$$

We evaluate what can be determined with the knowledge of at most $c - 2$ original data items. The only information known to the attacker:

$$X'_p = x_{p1}, x_{p2}, \ldots, U, \ldots, U, \ldots, x_{pc} \tag{5}$$

$$Y = y_1, y_2, \ldots, y_n \tag{6}$$

where X'_p is a set of $c - 2$ original data items, and each U represents a missing value. The goal of the attacker is to identif y two missing original values and determine which of these corresponds to the original value of y_t.

Case 1: There exist two items in the obfuscated dataset y_k, y_l that fall within the interval $[min(Y_p), max(Y_p)]$. In this case , the attacker knows that y_k, y_l are the missing items in the neighborhood p. These two items can be placed in the neighborhoo d in two ways, both of which produce the same obfuscated neighborhood Y_p:

$$X'_p = x_{p1}, x_{p2}, \ldots, y_k, \ldots, y_l, \ldots x_{pc} \tag{7}$$

$$X''_p = x_{p1}, x_{p2}, \ldots, y_l, \ldots, y_k, \ldots x_{pc} \tag{8}$$

Since there is no additional information that enables the attacker to accurately identify which of the two sequences X'_p, X''_p is the original neighborhood, the attacker cannot determine with certainty whether y_k or y_l is equal to x_t.

Case 2: There are no items in the obfuscated data set that fall within the interval $[min(Y_p), max(Y_p)]$. In this case, the missin g items are one of the three pairs: $min(Y_p) - 2, min(Y_p) - 1$, $max(Y_p) + 1, max(Y_p) + 2$ or $min(Y_p) - 1, max(Y_p) + 1$. For each pair, t here are two permutations of the neighborhood that could be the original neighborhood. In this case, the original value correspondi ng to y_t can be one of 6 values, and the attacker cannot determine with certainty which of these corresponds to x_t.

Case 3: One item in the obfuscated dataset lies in $[min(Y_p), max(Y_p)]$. Let this item be denoted as y_{kl}. In this case, the m issing items can be one of two pairs: $min(Y_p) - 1, y_{kl}$ or $y_k l, max(Y_p) + 1$. Each pair can fill up the missing positions in two ways. In this case, there are 4 candidates corresponding to the original value for y_t and again the attacker cannot know the value of x_t with certainty.

This shows that even with the knowledge of $c-2$ items in a neighborhood, the attacker cannot determine the original values of the remaining items with certainty.

□

The cluster preserving property of the linear geometric transformations make them attractive for use in DO, but their vulnerability to reversal makes them unsuitab le. The *NeNDS* transformation technique offers a stronger privacy preserving capability. In *GT-NeNDS*, The obfuscated dat a that results from Geometric transformations is obfuscated by *NeNDS*. Combining it with a stronger transformation function su ch as *NeNDS* strengthens the weak *reversibility* property of geometric transformations. The multi-tier obfuscation makes *GT-NeNDS* more difficult to reverse engineer than *NeNDS*. A comparison of the cluster retention capability is analyzed ex perimentally in Chapter 4, proving that *GT-NeNDS* is an optimum data obfuscation technique that provides robust d ata privacy as well as high data usability.

4. Experiment Results

The performance of the privacy framework using hybrid-NeNDS is discussed in this section. The experiments compare the prediction results of the obfuscated data with the prediction results of the original data. Several collaborative filtering approaches have been developed for recommendation systems. Automated CF systems are widely used for providing recommendations to users based on the ratings of users with similar interests. The different CF systems can be broadly classified into memory-based CF, model-based CF, and hybrid memory-model CF. Memory based systems use the raw data in the database by applying nearest neighbor techniques for predicting user preferences. Model-based approaches first create a model based on the available information and use this model to make probabilistic predictions for the unrated items. Hybrid approaches use the model based approach to create sets of similar users. The predictions are then made by using memory-based techniques on the set of similar users, thus optimizing the accuracy and time complexity of the predictions. In this paper, we use the Pearson's correlation co-efficient and Vector similarity algorithms, which are memory based approaches. We also use the personality diagnosis algorithm to analyze its performance on obfuscated data.

The experiment involves dividing the data (users and their ratings) into a training set and a test set. The training set is used as the database for the CF engine. Each user/ratings record in the test set is iteratively presented to the CF engine for making predictions. The ratings of the test user, known as the *active* user are divided into a set of observed ratings, I_a and a set of unrated ratings P_r. The ratings I_a are presented to the CF engine and the predicted ratings P_{CF} for the unrated items are compared with the set P_r.

The set of tests to compare the performance of one-at-a-time recommendations are measured by using the average absolute deviation of the predicted ratings p_i with respect to the actual ratings on items for which the test set users have entered ratings (r_i). This metric was first introduced in GroupLens [22] and is used as a standard for comparing CF systems. The mean absolute deviation for a single user on m_a predicted items is given by Equation 9. The error is averaged over all the users in the test set. Since the two data collections used here have different ranges for ratings, the normalized mean absolute error $NMAE$ is evaluated [7] as shown in Equation 10.

$$|\overline{E}| = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N} \tag{9}$$

$$|NMAE| = \frac{|\overline{E}|}{r_{max} - r_{min}} \tag{10}$$

The evaluation considers two different database collections. The BookCrossing [29] collection consists of three databases **[User-info, Book-info, and Ratings-info]**. The **User-info** database contains demographic information of $278,858$ users **[ID, Location, Age]**. The **[Book-info]** database has information regarding the title, ISBN, year of publication, author, publisher, and edition for $271,379$ books. The **[Ratings-info]** database contains a total of $1,149,780$ ratings by the listed users for the books specified in the database. The fields that are obfuscated are: **[User-info: Age]** and all the fields in the **[Ratings-info]** database. The second database collection, Movielens [8] consists of three databases **[User-info, Movie-info, and Ratings-info]**. The **User-info** database contains demographic information of 6040 users **[ID, Age, gender, occupation, zip]**. The **[Movie-info]** database has information regarding the Movie-ID, title, release date, and video release date for $3,900$ movies. The **[Ratings-info]** database contains a total of $2,811,983$ ratings by the listed users for the movies specified in the database. The fields that are obfuscated are: **[User-info: Age, zip code]**. The gender and occupation fields are removed from the database before sending it to the CF server. All the fields in the **[Ratings-info]** database are obfuscated.

4.1. Results

To evaluate the performance of the CF engine, we carried out three types of tests for the one-at-a-time and ordered-list recommendations. The All-but-one test provides all the ratings except one for each active user in the test set. The accuracy of prediction of the single rating is measured in this test. In the Given-2 test, the observed ratings set I_a contains only two ratings. The accuracy of predictions of the rest of the ratings in the unrated set P_r is analyzed here. Given-10 measures the accuracy of the predictions with 10 ratings in the active user's observed-ratings set.

The data collections are arbitrarily divided into three sets, each set representing the repository of one meta-store front. All three repositories are first obfuscated using NeNDS, where each data set was divided into 100 neighborhoods. All three repositories apply the same geometric transformation to the data. For the User-info data, a scaling transformation of 0.8 was applied for each field. The ratings-info database was transformed with a different scaling vector that was generated randomly. The resulting databases were then appended to form a single collection. The collection was then divided into a training set and test set in the ratio $75\% : 25\%$. Each of the entries in the test set is then added to the training set one at a time to determine the mean absolute error and ranking score for all three tests. The tests are performed three times, once with each CF algorithm: Pearson, Vector Similarity, and Personality diagnosis.

Table 4 shows the prediction results of the three algorithms for the 'All-but-1' case. The results obtained for the individual algorithms with original data match the results obtained in [11]. The normalized mean absolute error for the obfuscate data are consistent with the results for the original data. This shows that the privacy framework does not affect the CF for the all-but-1 case.

Table 5 contains the results for the Given-10 test. The results in this test indicate that the errors introduced in this case are much smaller than the errors introduced when only two ratings were provided to the CF engine. Two of the three algorithms yield similar results with and without data obfuscation. The performance of the algorithms with increasing number of Given ratings was evaluated. The error difference between the original and obfuscated results decrease exponentially with the increase in number of Given ratings.

The selection of a distribution range for random perturbation is a critical factor that affects the privacy and usability of data. The only input parameter for NeNDS is the neighborhood size NH. In [18], the sensitivity of the neighborhood size NH on the *Misclassification error* (MCE) in clustering was evaluated. Experiments were also conducted to evaluate the effect that the variation of the number of neighborhoods has on the prediction accuracy [17]. The resuts show that the performance of NeNDS is insensitive to the parameter NH. The ability of hybrid-NeNDS to provide privacy without trading off usability of the CF system makes it an excellent candidate for privacy protection of data used for CF.

Table 4. Prediction accuracy: All-but-one test

CF Algorithm	Orig. Data	Obf. Data	Error %
Pearsons Movielens	0.198	0.198	0.0
V. Similarity Movielens	0.241	0.242	0.1
P. Diagnosis Movielens	0.192	0.193	0.1
Pearsons Bookcrossing	0.201	0.202	0.1
V. Similarity Bookcrossing	0.211	0.211	0.1
P. Diagnosis Bookcrossing	0.201	0.203	0.2

Table 5. Prediction accuracy: Given-10 test

CF Algorithm	Orig. Data	Obf. Data	Error %
Pearsons Movielens	0.199	0.200	0.1
V. Similarity Movielens	0.208	0.209	0.1
P. Diagnosis Movielens	0.196	0.196	0.0
Pearsons Bookcrossing	0.201	0.202	0.1
V. Similarity Bookcrossing	0.237	0.239	0.2
P. Diagnosis Bookcrossing	0.197	0.201	0.4

5. Conclusion

This paper proposes a privacy preserving framework for collaborative filtering applications. While there has been tremendous growth in the areas of information retrieval and optimization measures for CF systems, there has been little research in the area of privacy preserving CF. Trust-based systems have been proposed to thwart targeted attacks on CF systems to promote or demote items maliciously. CF using factor analysis proposes a secure method for CF among

peers. This method can only be used among a known set of users, where an active user seeks out information. This paper proposes a privacy framework that allows automated recommendations to be made to users in a privacy preserving manner that ensures the privacy of users. The framework can be used to share information among multiple meta-store fronts for information for mutual gain. New sellers suffer an initial setback, referred to as cold-start, because of the lack of a data pool to provide recommendations to its users. The cold start problem can be averted by the presence of a shared CF engine. The experimental results indicate that the accuracy of CF engines remains nearly the same in spite of the preliminary data obfuscation process. Although the rank scoring metric indicated that the utility of the ranking order is decreased by data obfuscation, the error is only about 5% on average, which is an acceptable trade-off, given the benefits of a robust privacy-preservation mechanism.

References

[1] R. Agrawal and S. Ramakrishnan. "Privacy-Preserving Data Mining". In *ACM Special Interest Group on Management of Data*, pages 439–450, 2000.

[2] J. Canny. Collaborative filtering with privacy, 2002.

[3] J. Canny. "Collaborative Filtering with Privacy via Factor Analysis". In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 238–245, Tampere,Finland, Aug 2002.

[4] Y.-H. Chien and E. I. George. A bayesian model for collaborative filtering. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, San Francisco, California, 1999. Morgan Kaufmann.

[5] L. Cranor, J. Reagle, and M. Ackerman. Beyond concern: Understanding net users attitudes about online privacy, 1999.

[6] D. Goldberg, D. Nichols, B. Oki, and D. Terry. "Using Collaborative Filtering to Weave an Information Tapestry". *Communications of the ACM*, 35(12):61–70, Dec 1992.

[7] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

[8] Grouplens. "http://www.grouplens.org/data/".

[9] J. Herlocker, J. Konstan, A. Borchers, and J. Reidl. "An Algorithmic Framework for Collaborative Filtering". In *ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, Aug 2002.

[10] W. Hill, L. Stead, M. Rosenstein, and G. W. Fumas. "recommending and evaluating choices in a virtual community of use". In *ACM Conference on human factos in computer systems CHI'95*, pages 194–201, Denver, Colorado, 1995.

[11] B. J.S, H. D., and K. C. Emperical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[12] S. Lam and J. Riedl. Shilling recommender systems for fun and profit.

[13] P. Massa and P. Avesani. Trust-aware collaborative filtering for recommender systems, 2004.

[14] P. Massa and B. Bhattacharjee. Using trust in recommender systems: an experimental analysis, 2004.

[15] S. Oliveira and O. Zaane. "Privacy Preserving Clustering by Data Transformation". In *Proc. of the 18th Brazilian Symposium on Databases*, pages 304–318, Manaus, Brazil, Oct 2003.

[16] S. Oliveira and O. Zaane. "Achieving Privacy Preservation When Sharing Data for Clustering". In *Workshop on Secure Data Management in conjunction with VLDB2004*, Toronto, Canada, Aug 2004. Springer Verlag LNCS 3178.

[17] R. Parameswaran. *A Robust Data Obfuscation Approach for Privacy Preserving Collaborative Filtering*. PhD thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, May 2006.

[18] R. Parameswaran and D. Blough. A Robust Data-obfuscation Approach for Privacy Preservation of Clustered Data. In *Workshop on Privacy and Security aspects in Data Mining held in conjunction with the 2005 IEEE International Conference on Data Mining*, pages 18–25, Houston, Texas, 2005. IEEE.

[19] D. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles. Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, UAI 2000*, pages 473–480, Stanford, CA, 2000.

[20] H. Polat and W. Du. Privacy-preserving collaborative filtering using randomized perturbation techniques, 2003.

[21] S. P. Reiss. "Practical Data-swapping The First Steps". In *ACM Transactions on Database Systems*, volume 9, pages 20–37, Mar 1984.

[22] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.

[23] P. Resnick and H. R. Varian. Recommender Systems. In *Communications of the ACM*, volume 4, pages 56–58. ACM, 1997.

[24] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217, 1995.

[25] L. Sweeney. "k-Anonymity: A Model for Protecting Privacy". *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[26] L. Ungar and D. Foster. Clustering methods for collaborative filtering. In *Proceedings of the Workshop on Recommendation Systems*. AAAI Press, Menlo Park California, 1998.

[27] S. Upendra. Social information filtering for music recommendation, 1994.

[28] C. D. Wolfgang. Preventing shilling attacks in online recommender systems paul-alexandru chirita.

[29] C.-N. Ziegler and D. Freiburg. "http://www.informatik.uni-freiburg.de/ cziegler/BX/".