Trading Robustness for Privacy in Decentralized Recommender Systems

Zunping Cheng and **Neil Hurley**
Information Hiding Laboratory
School of Computer Science and Informatics
University College Dublin, Ireland
{zunping.cheng, neil.hurley}@ucd.ie

Abstract

Collaborative filtering (CF) recommender systems are very popular and successful in commercial application fields. One end-user concern is the privacy of the personal data required by such systems in order to make personalized recommendations. Recently, peer-to-peer decentralized architectures have been proposed to address this privacy issue. On the other hand system managers must be concerned about system robustness. In particular, it has been shown that recommender systems are vulnerable to profile injection, although model-based CF algorithms show greater stability against malicious attacks that have been studied in the state-of-the-art. In this paper we generalize the generic model for decentralized recommendation and discuss the trade-off between robustness and privacy. In this context, we argue that exposing knowledge of the model parameters allows new, highly effective, model-based attack strategies to be considered. We conclude that the security concerns of privacy and robustness stand in opposition to each other and are difficult to satisfy simultaneously.

Introduction

Recommender systems use automated recommendation algorithms, such as collaborative filtering (CF), to help people discover what they need in a large set of alternatives by analyzing the preferences of other related users. With the rapid proliferation of online businesses, such systems are playing a more and more important role in web-based commercial operations and are attracting more and more users. A recent survey (A|Razorfish 2007) reports that 62% of investigated consumers have made a purchase based on personalized recommendations and 72% of them show great interest in purchasing goods with the help of recommendation engines.

However, as discussed for example in (Lam, Frankowski, and Riedl 2006), serious privacy issues emerge in recommender systems. Many ideas have been proposed to address privacy concerns. Among them, a popular approach is to decentralize the data management. Traditional recommendation architectures use centralized data management for a single application domain and lack direct interaction between users in the recommendation process. Peer-to-peer (P2P) architectures allow unstructured connectivity between peers in a network among whom the system data is distributed. In P2P-based recommender systems, there are no conventional dominant data servers, recommendation logic and users' rating data are spread among all the peers. From a scalability point-of-view this implies that there is no communication bottleneck to a single point. P2P also has a potentially positive side-effect which is that it is able to preserve users' privacy since personal information is kept within peers (Canny 2002).

Robustness of recommendation algorithms has been studied for several years, in the context of profile injection or shilling attacks (O'Mahony et al. 2004; Lam and Riedl 2004). In such attacks, malicious end-users, motivated to modify the recommendation output of the system, create false user profiles (sometimes called sybils), to distort the recommendation process. As an example of such an attack, a user, motivated to promote the rating of a product in order to boost its sales, might create a set of false profiles that rate that product highly (a so-called *push attack* (O'Mahony et al. 2004)). Many different attack strategies have been studied and they have been categorized in (Mobasher, Burke, and Sandvig 2006) as Sampling attacks, Random attacks, Average attacks, Bandwagon attacks and Segment attacks. Among these, the Average attack is the most effective and the Bandwagon attack needs the least knowledge. In (Mobasher, Burke, and Sandvig 2006; Jeff J. Sandvig 2008) empirical studies show that model-based CF algorithms are much more robust against these attack categories in comparison to memory-based algorithms.

In this paper, we argue that the set of attack categories that have been considered to date is incomplete. In particular, we argue that there is a need to consider *informed model-based* attacks, that is, attacks that use knowledge of the underlying algorithm. These attacks present a new vulnerability for model-based algorithms that has not been considered in previous work. In particular, we will show how such attacks can be applied very effectively against P2P recommendation algorithms. The contributions we make are as follows:

- **Model-based attacks** Beyond existing attack strategies, we propose to explore *informed model-based* attack strategies applied to model-based recommendation algorithms. Experiments show that with full knowledge or limited knowledge, theses attacks outperform strategies proposed previously.

- **Tradeoff between robustness and privacy** P2P-based systems aim to protect users' privacy by allowing the exchange of the algorithm parameters among users, rather than raw personal ratings. However, because those parameters contain very important meta information, attackers can take advantage of them in constructing attacks. Thus along with privacy enhancement, the whole system becomes more vulnerable and a tradeoff between robustness and privacy arises.

Motivation and Context

Decentralized Recommendation with Privacy

Privacy preservation is one of the main motivations of decentralized recommender systems. (Canny 2002) was the first to propose a P2P-based architecture for privacy preservation which was implemented in the Mender system. This work has been followed by (Miller, Konstan, and Riedl 2004), which describes P2P-based CF algorithm using item-item similarity and includes five different architectures for locating neighbors for the model. (Berkovsky et al. 2007) makes use of data modification techniques to mitigate some privacy issues in a decentralized environment. It also discusses the tradeoff between accuracy and privacy in CF systems. (Aïmeur et al. 2008) proposes a hybrid system called ALAMBIC, which protects user privacy information by introducing a third party between users and merchant. In (Lam, Frankowski, and Riedl 2006) security and privacy issues in recommender systems are discussed, highlighting three concerns: the value of and risk to users' shared information, the effectiveness of malicious attacks and the issues involved in constructing P2P recommenders.

Model-based Collaborative Filtering Algorithms

In model-based CF algorithms, a theoretical model is proposed of user rating behavior. Rather than use the raw rating data directly in making predictions, instead the parameters of the model are estimated from the available rating data and the fitted model is used to make predictions, see Figure 1. Many model-based CF algorithms have been studied over the last ten years. For example, (Breese, Heckerman, and Kadie 1998) discusses two probabilistic models, namely, clustering and Bayesian networks. In (O'Connor and Herlocker 1999), four partitioning-based clustering algorithms are used to make predictions, leading to better scalability and accuracy in comparison to random partitioning. The probabilistic latent semantic analysis (PLSA) algorithm is introduced to CF recommendation in (Hofmann 2004). Its main idea is to employ latent class variables to learn users' communities and valuable profiles and then make predictions based on them. In (Canny 2002), the EM algorithm is used to train a linear factor analysis model.

Informed Model-based Attack Strategies

The possibility of biasing a recommender system's rating output by the creation of false profiles was first raised in (O'Mahony et al. 2004). Since then, a classification of such *profile injection* attacks has been proposed in (Burke, Mobasher, and Bhaumik 2005) and the effectiveness of such attacks has been evaluated on both memory-based and model-based recommendation algorithms (Mobasher, Burke, and Sandvig 2006; Jeff J. Sandvig 2008). The five general attack strategies proposed in (Mobasher, Burke, and Sandvig 2006) are Sampling attacks, Random attacks, Average attacks, Bandwagon attacks and Segment attacks. In practice, an Average attack is much more effective than a Random attack. The Bandwagon attack is nearly as effective as the Average attack. Random, Average, and Bandwagon attack do not work well against item-based collaborative filtering

However, we contend that this analysis is incomplete and in particular lacking in two aspects:

1. All proposed attacks are *uninformed* in the sense that they do not take explicit account of the system recommendation algorithm, although some account is taken of the level of knowledge of the statistics of the rating dataset.
2. They do not pay sufficient attention to the level of *perceptibility* of the attacks – that is, to what extent are attack profiles distinguishable from genuine profiles.

It might be argued that potential attackers do not have detailed knowledge of the underlying algorithm and thus 1. is an unreasonable assumption. However, this is a flawed argument that has been rejected in mainstream security research. Kerckhoff's principle from cryptography, for instance, states that a system must be secure under the conditions that the attacker knows *everything* about the underlying algorithm, except the secret keys. As quoted from (Mann 2002): "*Kerckhoffs' principle applies beyond codes and ciphers to security systems in general: every secret creates a potential failure point. Secrecy, in other words, is a prime cause of brittlenessłand therefore something likely to make a system prone to catastrophic collapse. Conversely, openness provides ductility.*" Indeed, our previous work on memory-based CF algorithms (O'Mahony et al. 2004), has developed the most effective profile-injection attacks by explicitly exploiting weaknesses in the underlying algorithm. Work such as (Burke, Mobasher, and Bhaumik 2005; O'Mahony et al. 2004) has examined how much statistical knowledge of the rating database is needed to launch successful attacks and has shown that effective attacks can be launched with knowledge of the statistics of the most popular items. However, this work has *not* considered exploiting knowledge of the underlying recommendation algorithm.

In the case of 2., although various supervised and unsupervised classification algorithms have been tested against the proposed attack types (Burke, Mobasher, and Bhaumik 2005; Mobasher, Burke, and Sandvig 2006; Jeff J. Sandvig 2008), the question of whether attackers can explicitly develop statistically imperceptible but effective attacks has not yet been fully explored. In particular, for model-based algorithms, it is interesting to ask, to what extent can the model parameters be estimated by a third-party interacting with the system and how might the security of the system be compromised by this. Such information leakage has been addressed recently in the context of the security of watermarking algorithms (Comesaña, Pérez-Freire, and Pérez-González 2005).

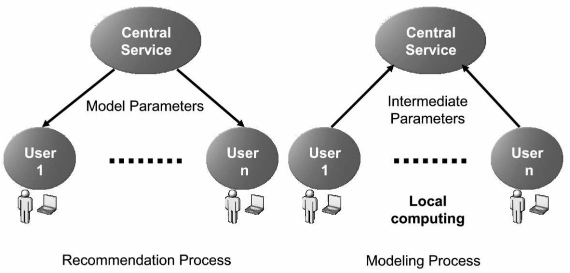

Figure 1: P2P-based Recommendation Framework

It is easy to imagine that parameter leakage might impact on data privacy, which compromises the end-user's security, as well as robustness, which compromises the overall system security. For this reason, in this paper we explore the relationship between robustness and privacy, in the context of a peer-to-peer CF algorithm, developed explicitly with end-user privacy in mind (Canny 2002). In fact, the approach taken in (Canny 2002), is to make the system parameters freely available to all end-users, in order to allow users to keep their personal rating data secure. As we will show, access to the system parameters is a security hole that can be effectively exploited by profile injection attacks.

The Mender Recommender System

P2P Architecture

In the recommendation architecture (called Mender) proposed by (Canny 2002) a P2P network of user peers collaborates to compute the recommendation model parameters and the predicted ratings for each user. The architecture contains two types of peers. One is a central service (called a *totaller* in Canny's paper). The other is a common user.

In Figure 1 the two phases of the P2P-based CF system are shown: modeling and recommendation. During modeling, central service peers collect intermediate parameters from user peers and calculate final model parameters. While in recommendation, user peers collect model parameters from the central services and then predicted ratings are computed locally using the user's own rating data. From a privacy perspective, users never need to divulge their ratings to other users. Instead, the P2P system, computes and distributes the model parameters. As well as dealing with privacy, Canny's model also takes account of the fact that some totallers may not be trustworthy. The distributed algorithm for computing model parameters has been shown to be robust even when a certain percentage of totallers do not perform their task correctly. However, Canny does not analyze the security risk to the system if attack sybils are present.

Factor Analysis Model

In (Canny 2002) a linear model is proposed to describe the user rating process. In the model, it is assumed that there exist some set of underlying hidden categories and that a user's preference for a particular item is a linear combination of how well the item fits into each category and how much the user likes each category. Specifically, the linear model for user preferences is as follows: Let n be the total number of items and m the total number of users. Let the full set of items be denoted by $\mathcal{I}$ and for an item $i \in \mathcal{I}$, let $\mathcal{R}$ be the set of all users, and R_i be the set of users that have rated i. Then

$$\mathrm{Y} = \Lambda^T \mathrm{X} + \mathrm{N}\,, \tag{1}$$

where Y is the $n \times m$ rating matrix, $\mathrm{Y} \triangleq (\mathbf{y}_1, \ldots, \mathbf{y}_m)$ and $\mathbf{y}_u$ is the n-dimensional vector of ratings for user u over all items; $\mathrm{X} \triangleq (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ is an $k \times m$ matrix of user preferences for $k < n$ hidden categories where $\mathbf{x}_u$ is the k-dimensional preference vector for user u. $\Lambda \triangleq (\boldsymbol{\lambda}_1, \ldots \boldsymbol{\lambda}_n)$ is the $k \times n$ matrix, with column vectors $\boldsymbol{\lambda}_i$, representing the extent to which each item fits in each category. The $n \times m$ matrix N represents noise in the rating process. The rows of X are assumed to be iid random variables drawn from $\mathcal{N}(0, 1)$. The noise is also assumed to consist of zero mean gaussian iid random variables with a fixed variance ψ. The parameters of the model required in order to make predictions are the matrix Λ and the noise variance ψ.

In order to understand how to build an informed model-based attack on this system, we must describe the steps involved in learning the system parameters from a given rating matrix Y. As both X and Λ are unknown, the task is find a factorization of the ratings matrix into the product of two such matrices. Canny uses the iterative expectation maximization (EM) algorithm to carry out the factorization of the highly sparse matrix Y. It is most useful to describe this algorithm from the perspective of a particular item i. Before applying the algorithm, the items means are removed from the rating matrix, so that, in the following $\mathbf{y}_i$ are zero mean ratings. Using the superscript (l) to represent the value of a variable at the l^{th} iteration, the algorithm iteratively calculates

$$\begin{aligned} \mathrm{B}_i^{(l)} &= \sum_{u \in R_i} \left(\mathbf{x}_u^{(l)} \mathbf{x}_u^{(l)^T} + \psi^{(l)} \mathrm{M}_u^{(l)} \right) \\ \mathbf{b}_i^{(l)} &= \sum_{u \in R_i} \frac{y(u,i)}{n_u} \mathbf{x}_u^{(l)} \end{aligned} \tag{2}$$

where

$$\mathrm{M}_u^{(l)} = \left(\psi^{(l)} \mathrm{I} + \Lambda^{(l)} \mathrm{D}_u \Lambda^{(l)^T} \right)^{-1}, \tag{3}$$

the matrix D_u is the $n \times n$ diagonal matrix with 1 in position (j, j) where j is an item that user u has rated and zero otherwise and n_u is the number of items rated by u. Then, the i^{th} column of Λ^l and ψ are updated by

$$\begin{aligned} \boldsymbol{\lambda}_i^{(l+1)} &= \mathrm{B}_i^{(l)^{-1}} \mathbf{b}_i^{(l)} \\ \psi^{(l+1)} &= \frac{1}{m} \sum_i \left(\sum_{u \in R_i} \left(\frac{y(u,i)^2}{n_u} \right) - \boldsymbol{\lambda}_i^{(l+1)^T} \mathbf{b}_i^{(l)} \right). \end{aligned} \tag{4}$$

As $l \to \infty$, $\Lambda^{(l)}$ and $\psi^{(l)}$ converge to Λ and ψ, respectively. Then, a user with rating vector $\mathbf{y}_u$, using Λ and ψ, makes a prediction $p(u, i)$ for a given user item pair is given by

$$\mathbf{x}_u = \mathrm{M}_u \Lambda \mathrm{D}_u \mathbf{y}_u \tag{5}$$

$$p(u, i) = \boldsymbol{\lambda}_i^T \mathbf{x}_u\,. \tag{6}$$

An Informed Model-based Attack

In the following, we will first consider how Canny's system can be attacked if *full knowledge* of the system is available to the attacker, *including* the ratings database. While this is an unrealistic scenario for most real-world systems, it provides an upper bound on the worst-case attack that can be applied to the system. We note in passing however, that some real-world online systems *do* indeed expose the rating database to end-users (e.g. `www.mouthshut.com` and `www.xstreetsl.com`). Access to the full ratings database allows exact computation of the factor matrix X, which in reality does not need to be exposed for prediction, as the required part can be directly computed from (5). However, as we will show, estimating X is sufficient to create an effective attack, using *only* the public parameters.

With Full Knowledge

Let i represent an item that is to be pushed by the creation of false profiles. Examining (6), we note that the predictions for i depend on the i^{th} column of Λ and that the prediction will be maximized for a user u by selecting $\boldsymbol{\lambda}_i = \alpha \mathbf{x}_u$, for some α. It follows also that to maximize the average prediction shift over all users that have not rated i, the optimal $\boldsymbol{\lambda}_i$ should be chosen as

$$\boldsymbol{\lambda}_i^* = \alpha \sum_{u \in \mathcal{R} - R_i} \mathbf{x}_u \, . \tag{7}$$

Note from (5) that $\mathbf{x}_u$ is itself dependent on $\boldsymbol{\lambda}_i$. Nevertheless, iterating (7) and (5) converges quickly to a fixed optimal $\boldsymbol{\lambda}_i^*$.

Consider the insertion of a single attack profile, $\mathbf{y}_a$. The goal of the attacker may now be described as to select $\mathbf{y}_a$, so that the EM algorithm, using the dataset augmented with $\mathbf{y}_a$ converges to a matrix Λ with i^{th} column given by $\boldsymbol{\lambda}_i^*$. The first step is to select the profile size, n_a and the set of items from which the profile will be constructed. n_a is selected as a typical size of a user profile and the consistent items are selected at random.

Next note that, with the additional attack profile, (2), is modified as $\mathrm{B}_i' = \mathrm{B}_i + \mathrm{H}_i$ and $\mathbf{b}_i' = \mathbf{b}_i + \mathbf{h}_i$ where

$$\mathrm{H}_i \triangleq \frac{1}{n_a}(\mathbf{x}_a \mathbf{x}_a^T + \psi \mathrm{M}_a) \tag{8}$$

$$\mathbf{h}_i \triangleq \frac{y(a,i)}{n_a} \mathbf{x}_a \tag{9}$$

and hence, from (4),

$$\boldsymbol{\lambda}_i^{\text{new}}(\mathbf{x}_a) = (\mathrm{B}_i + \mathrm{H}_i)^{-1}(\mathbf{b}_i + \mathbf{h}_i) \, . \tag{10}$$

where on the lhs we have explicitly represented the dependence of $\boldsymbol{\lambda}_i^{\text{new}}$ on $\mathbf{x}_a$. Thus, we define an optimization problem, for the optimal $\mathbf{x}_a$:

$$\mathbf{x}_a^* \triangleq \arg\max_{\mathbf{x}} \boldsymbol{\lambda}_i^{*T} \boldsymbol{\lambda}_i^{\text{new}}(\mathbf{x}) \tag{11}$$

We note that (11) is an unconstrained multidimensional nonlinear maximization problem, which we solve using the Nelder-Mead simplex method, implemented in Matlab. The profile values $\mathbf{y}_a$ are next selected, given $\mathbf{x}_a^*$. Using (5) we choose

$$\mathbf{y}_a^* = (\mathrm{M}_a \Lambda \mathrm{D}_a)^{\dagger} \mathbf{x}_a^* \, ,$$

where $\mathrm{A}^{\dagger}$ is the pseudo-inverse of a matrix A. In a final step, the item means are added back into the attack profile and the ratings are truncated and quantized to the rating scale.

As changes to any single column of Λ can impact on the values of the other entries in the matrix and these changes are not explicitly accounted for in our attack algorithm, to avoid large error propagation, attack profiles are added one-by-one and after each addition, the full EM algorithm is run to recalculate Λ and ψ.

With Limited Knowledge

Although the attack algorithm outlined above appears to depend intimately on the actual ratings in the database, Λ and ψ are learned on the model assumption that the hidden variables X are iid gaussian variables. Working backwards, given Λ and ψ, we generate a random matrix drawn from $\mathcal{N}(0,1)$ and then generate a ratings database using

$$\mathrm{Y} = \Lambda^T \mathrm{X} \, .$$

We now apply the attack described in the previous section, using the synthetic dataset and the parameters which were generated from the real data. As shown later, this turns out to be a very effective attack, out-performing the Average attack, the most effective of the non-informed attacks categorized in (Mobasher, Burke, and Sandvig 2006). Although the Average attack does not use explicit knowledge of the recommendation algorithm, it does depend on a high level of statistical knowledge about the rating dataset. Other than the public parameters, the only other information used in our informed attack is an estimate of item mean, which is a single value applied to all items. In contrast, the Average attack requires the item mean and standard deviation of *all* items in the dataset.

Experiments

In order to examine the performance of our model-based FA attack, five attacks are selected to test. FA attack means FA model-based attack with full knowledge, that is the whole rating data. FA-x attack means FA model-based attack with only the system parameters. Random, Average and Bandwagon attacks are designed according to (Burke, Mobasher, and Bhaumik 2005).

Evaluation Metrics

The prediction shift (PS) metric was introduced by (O'Mahony et al. 2004) and used by (Mobasher, Burke, and Sandvig 2006). It measures the effectiveness of an attack by the difference between predictions before and after the attack. PS can be defined as follows:

$$E_p(u,j) = p'(u,j) - p(u,j) \tag{12}$$

where $E_p(u,j)$ represents the prediction shift for user u on item i, p_j' and p_j are post- and pre-attack predictions respectively. We also introduce a new metric to evaluate the attack:

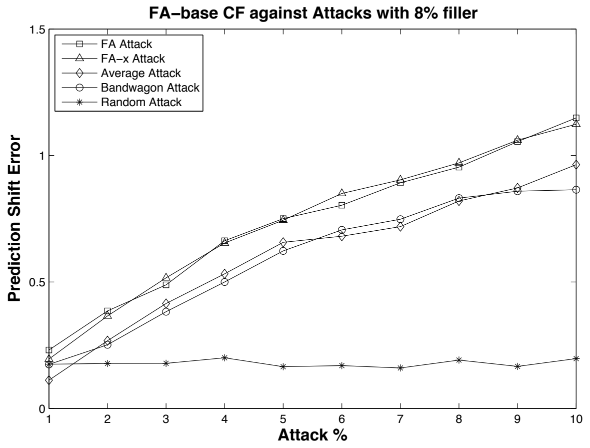

Figure 2: Prediction Shift(PS): FA, FA-x, Average, Bandwagon, and Random Attacks against FA-based CF algorithm.

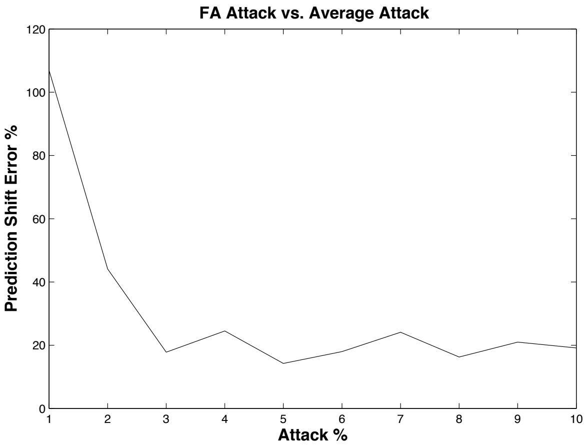

Figure 3: Prediction Shift(PS): FA Attack vs. Average Attack against FA-based CF algorithm.

high rating ratio (HRR), which shows how much predictions are pushed to high values for an attacked item.

$$H(r, j) = \frac{|u \in U|p'(u, j) \geq r|}{|u \in U|p(u, j) \geq r|} - 1 \tag{13}$$

where j is the attack item, r is the given rating threshold, and $U \in R$ is a subset of users for which predictions are made.

Data and Test Sets

The larger data set of MovieLens is adopted in our experiments, which consists of approximately 1 million ratings for 3952 movies by 6040 users. Movies are rated on a scale of one to five. From this dataset, we extract a series of subsets to conduct our tests. Each of them consists of 1220 items. The average sparsity of the selected rating matrices is 10.31%.

Evaluation of Informed Model-based Attack

To evaluate the attack, we take the following approach. A subset of 200 users is extracted from the Movielens dataset along with 1220 items, which were rated by three or more users. The dataset is divided randomly in a 50:50 ratio into training and test sets, consisting of 100 users each. The model parameters are learned by applying Canny's algorithm to the training set. An item is selected at random on which to apply a push attack. Predictions are made for the attack item for users in the test set. False profiles are then injected into the training set and the parameters re-learned. Predictions are made for the users in the test set and the prediction shift over all users in the test set is calculated. The process of profile injection and prediction shift calculation is repeated 50 times. The average of the 50×100 prediction shifts is calculated as the attack performance.

We evaluate the results based on two parameters: attack size and filler size. Attack size means the percentage of the number of attack profiles against the size of the pre-attack training set. Filler size is the percentage of items rated by attackers against the total number of items. For all tests we select 10% as filler size and from 1% to 10% as attack size. Figure 2 shows FA and FA-x attacks outperform Random, Average and Bandwagon attacks based on the PS metric. The Average attack is similar to the Bandwagon attack. The Random attack is worst against FA-based CF and the prediction shift is very low. This validates the results from (Mobasher, Burke, and Sandvig 2006; Jeff J. Sandvig 2008): model-based CF algorithms are robust to these simple attacks. However, just as expected, the informed attack is most successful. From Figure 3 we observe that the FA attack is about 20% better than the Average attack when the attack size is $> 3\%$. The HRR results in Figure 4 show that the FA attack is $> 60\%$ better than the Average attack. We find that although the FA-x attack has almost the same performance as FA attack by the PS metric, the former is obviously not as good as the latter by the HRR metric.

Conclusions

While robustness and privacy are two key issues for recommender systems, our analysis has shown that improving one can have a detrimental effect on the other – exposing system parameters in order to protect raw user data, gives malicious users a new means of attacking the system. We note however that explicit exposure of the parameters is not necessary for the construction of informed model-based attack strategies – indeed it is possible to estimate these parameters from normal user interactions, even if they are not made available by the system. In future work, we will explore the effectiveness of informed model-based attacks on other model-based systems, (e.g. k-means and PLSA) and expect that this will lead to much more effective attacks than previously applied. We will also explore the issue of attack perceptibility. One advantage of informed model-based attacks (from the attackers

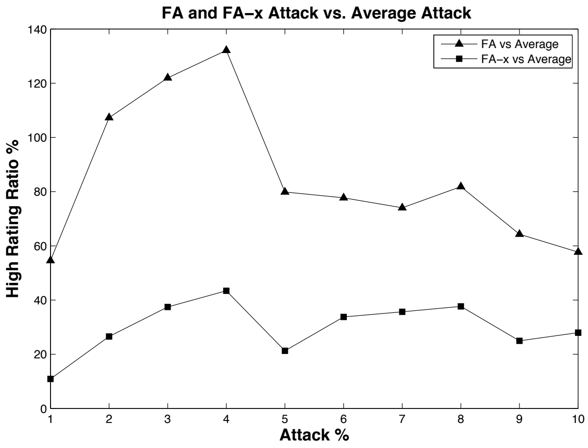

Figure 4: High Rating Ratio(HRR): FA and FA-x Attack vs. Average Attack against FA-based CF algorithm, r = 4.

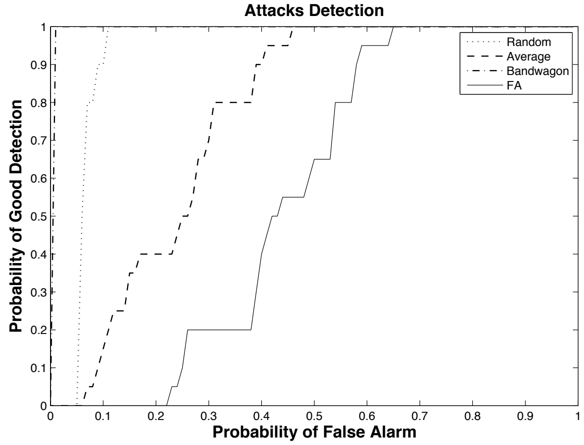

Figure 5: Model-based Detection.

point-of-view) is that false profiles constructed in this manner are statistically very similar to genuine profiles. Figure 5 shows the detection results, presented as a ROC curve of probability of false alarm against probability of good detection, for a particular statistical detector applied to various different types of attack profiles. From it, we can see that the FA attack is the lease detectable and the Bandwagon attack the most detectable. The issue of attack detection will be discussed in detail in a future paper.

Acknowledgements This work is supported by Science Foundation Ireland, grant number 07/RFP/CMSF219.

References

Aïmeur, E.; Brassard, G.; Fernandez, J. M.; and Onana, F. S. M. 2008. Alambic: a privacy-preserving recommender system for electronic commerce. *International Journal of Information Security* 7(5):307–334.

A|Razorfish, A. 2007. Digital consumer behavior study. In *http://www.razorfish.com/reports/DigConsStudy.pdf.*

Berkovsky, S.; Eytani, Y.; Kuflik, T.; and Ricci, F. 2007. Enhancing privacy and preserving accuracy of a distributed collaborative filtering. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, 9–16. ACM.

Breese, J. S.; Heckerman, D.; and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, 43–52. UAI.

Burke, R.; Mobasher, B.; and Bhaumik, R. 2005. Limited knowledge shilling attacks in collaborative filtering systems. In *Proceedings of the 3rd IJCAI Workshop in Intelligent Techniques for Personalization*. IJCAI.

Canny, J. 2002. Collaborative filtering with privacy via factor analysis. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 238–245. ACM.

Comesaña, P.; Pérez-Freire, L.; and Pérez-González, F. 2005. Fundamentals of data hiding security and their application to spread-spectrum analysis. In Barni, M.; Herrera-Joancomartí, J.; Katzenbeisser, S.; and Pérez-González, F., eds., *Information Hiding*, volume 3727 of *Lecture Notes in Computer Science*, 146–160. Springer.

Hofmann, T. 2004. Latent semantic models for collaborative filtering. *ACM Transactions on Internet Technology* 22(1):89–115.

Jeff J. Sandvig, Bamshad Mobasher, R. D. B. 2008. A survey of collaborative recommendation and the robustness of model-based algorithms. *IEEE Data Engineering Bulletin* 31(2):3–13.

Lam, S. T. K., and Riedl, J. 2004. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web*, 393–402. ACM.

Lam, S. K.; Frankowski, D.; and Riedl, J. 2006. Do you trust your recommendations? an exploration of security and privacy issues in recommender systems. In *Proceedings of Emerging Trends in Information and Communication Security, International Conference, ETRICS 2006*, 14–29. LNCS.

Mann, C. C. 2002. Homeland insecurity. *The Atlantic Monthly* 290(2).

Miller, B. N.; Konstan, J. A.; and Riedl, J. 2004. Pocketlens: Toward a personal recommender system. *ACM Transactions on Information Systems* 22(3):437–476.

Mobasher, B.; Burke, R.; and Sandvig, J. 2006. Model-based collaborative filtering as a defense against profile injection attacks. In *Proceedings of the 21st National Conference on Artificial Intelligence*. AAAI.

O'Connor, M., and Herlocker, J. 1999. Clustering items for collaborative filtering. In *Proceedings of the ACM SIGIR Workshop on Recommender Systems*. ACM.

O'Mahony, M.; Hurley, N.; Kushmerick, N.; and Silvestre, G. 2004. Collaborative recommendation: A robustness analysis. *ACM Transactions on Internet Technology* 4(4):344C377.