Collaborative Filtering Under a Sybil Attack: Analysis of a Privacy Threat

Davide Frey, Rachid Guerraoui, Anne-Marie Kermarrec, Antoine Rault

Collaborative Filtering Under a Sybil Attack: Analysis of a Privacy Threat

Davide Frey
INRIA Rennes
France
davide.frey@inria.fr

Rachid Guerraoui
EPFL
Switzerland
rachid.guerraoui@epfl.ch

Anne-Marie Kermarrec
INRIA Rennes
France
anne-marie.kermarrec@inria.fr

Antoine Rault
INRIA Rennes
France
antoine.rault@inria.fr

ABSTRACT

Recommenders have become a fundamental tool to navigate the huge amount of information available on the web. However, their ubiquitous presence comes with the risk of exposing sensitive user information. This paper explores this problem in the context of user-based collaborative filtering. We consider an active attacker equipped with externally available knowledge about the interests of users. The attacker creates fake identities based on this external knowledge and exploits the recommendations it receives to identify the items appreciated by a user. Our experiment on a real data trace shows that while the attack is effective, the inherent similarity between real users may be enough to protect at least part of their interests.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage And Retrieval—*Information filtering*

Keywords

Recommender,collaborative filtering,privacy,sybil attack

1. INTRODUCTION

Recommenders have become ubiquitous in a large number of online services. Introduced in the early 1990s, to deal with the information overload brought about by the internet [17], they quickly spread from initial specialized application such as the selection of Usenet news [31] to widely available commercial platforms like Amazon.com, which filed a patent for its recommender service in 1998 [25]. Today, recommenders span a wide range of websites including social networks [1, 3], news websites [16], and multimedia services [7, 2]; and they have even been proposed to help patients select the most suitable physicians [20].

Recommenders collect information about user opinions through the interactions between users and, most often, the websites they are deployed on. Such information can take an implicit or explicit form. Implicit information includes for example clicks on links, or the time spent on a web page or listening to a song. Explicit information includes ratings such as on Netflix and Amazon, comments and reviews, or application-specific actions such as likes on Facebook, or retweets on Twitter. In both cases, recommenders collect this information into user profiles that characterize the interests of users. They then use these profiles to attempt to predict the tastes of users on items they have not yet been exposed to.

Two major classes of recommenders exist: content-based, and collaborative-filtering. The former rely on the characteristics of the items appreciated by users to identify other items that are susceptible to interest them. The latter exploit the fact that users that appreciated similar items in the past will probably appreciate similar items in the future. They therefore identify items to recommend among the items appreciated by similar users. Collaborative-filtering comprises the vast majority of existing recommendation platforms thanks to its content-agnostic nature. Collaborative-filtering systems can recommend complex objects for which accessing and processing content would be difficult. This gives them the flexibility to operate with highly heterogeneous items (for example recommending paintings to a user based on their literary tastes).

Regardless of the techniques they use, the growing popularity and pervasiveness of recommenders have raised concerns from the research community and from users themselves about the threats they pose to user privacy. As the data exploited by recommenders gets more and more into the personal sphere, users have started to worry about their data being collected and concentrated in the data centers of a few big players, and potentially sold to or stolen by potentially malicious third parties. This concern for the emergence of online Big Brothers has prompted not only researchers but also user communities to design decentralized solutions for recommenders [14, 11].

However, the threat of a Big Brother is only one of the risks associated with the use of a recommender. Decentralized solutions replace the Big Brother by a plethora of little brothers among which it is easy to envision the presence of ill-intentioned users. But even in centralized ones, recent research has shown that the very nature of collaborative recommendation makes it reasonably easy for an attacker to learn about and monitor the interests of other users simply by observing publicly available information [13]. In this paper, we follow the research direction laid out by [13] and evaluate the impact of an active Sybil attack in which the

attacker attempts to guess the items contained in a target user's profile.

Our preliminary results on the MovieLens-100k dataset highlight a sharp difference with respect to what [13] shows for passive attacks. The attacker requires a significant amount of auxiliary knowledge to be successful: knowing 50% of the target's ratings only allows the attacker to identify other items with less than 50% of accuracy.

2. PROBLEM STATEMENT

We consider a recommender based on user-based collaborative filtering. As mentioned in Section 1, collaborative-filtering algorithms are particularly successful due to their flexibility. In the user-based variant, a collaborative-filtering system essentially consists of a k-nearest-neighbor (KNN) algorithm that identifies for each user, the set of most similar other users according to the opinions they expressed on the items they have been exposed to. In the following, we consider a system in which users rate items with a numerical score (e.g. 1 to 5). For each user, u, the system collects the mapping between items and scores in a user-profile data structure. It then runs a KNN algorithm to identify the users associated with the most similar user profiles. After identifying u's neighbors, the system ranks the items they have rated—for example through a combination of ratings, number of neighbors that rated them, and similarity of u with those neighbors—and recommends to u the top-ranking ones to which she has not yet been exposed.

We consider an attacker that has the capability to (i) observe part of the ratings expressed by the target—we refer to these ratings as auxiliary information—and (ii) create a number of fake identities (Sybils) with the objective to extract information from the recommender [13]. We also assume that the attacker knows the value of k, the size of neighborhoods in the KNN algorithm, which enables her to create the right number of Sybils.

Attackers can obtain auxiliary information in several ways depending on the characteristics of the target recommendation system. In a system like Amazon.com, the attacker can use reviews posted by the target as evidence that she bought this or that item, or posts on social networks such as "I just bought item X". In systems like Last.fm, the attacker may consult publicly available listening histories. In decentralized systems like [11], she may simply exploit the profile exchanges at the basis of the recommendation algorithm.

How to create fake identities also depends on the system being attacked. On some websites, it is enough to create a number of accounts and perform some actions such as listening to music tracks, or adding reviews. On others, it may be more complex as creating a profile might involve purchasing items or other costly actions. For the purpose of this paper, we assume that the attacker has the means to give each of its fake identities a profile consisting of the set of auxiliary items and associated ratings. In case creating a profile involves costly actions, the attacker may simply shrink the set of auxiliary items to a manageable number. If the set of ratings copied by the Sybils is large enough, the KNN algorithm should give each Sybil a neighborhood consisting of the target user and the remaining Sybils. The attacker can then monitor all the recommendations received by each of the Sybils and assume that all the recommended items come from the profile of the target.

Although [13] introduced this attack, it did not evaluate its effectiveness. Its authors only suggest that attacks based on auxiliary knowledge should be effective as long as the Sybils have access to $O(log(N))$ items, N being the number of users. In the following, we show that this is not the case with non-binary ratings in the dataset we considered.

3. EXPERIMENTAL PROTOCOL

We implemented the Sybil attack described above using the user-based collaborative-filtering implementation provided by Mahout 0.9 [4], a machine-learning library developed by the Apache Foundation. Mahout allows us to concentrate on the implementation of the attack, while using a well-tested implementation of state-of-the-art user-based collaborative filtering.

To compute the KNN graph, we adopt the well-known cosine-similarity metric. Like most other recommenders, Mahout uses a slightly modified version of the similarity which computes the norms of the two profiles by counting only the items that are common to both of them.

$$Cos(u,v) = \frac{u \cdot v}{\|u_v\|\|v_u\|},$$

where $x_y = \{(i,r) \in x | \exists r', (i,r') \in y\}$, and (i,r) refers to an item and its rating.

We ran our experiments on ML-100k, a dataset consisting of 100,000 movie ratings from the MovieLens [19] online movie recommender. Each rating consists of an integer value from 1 to 5, 1 being the worst, and 5 being the best.

We assume that the auxiliary information available to the attacker consists not only of a list of items, but also of the associated ratings as expressed by the target. We consider several percentages of auxiliary items available to the attacker. As discussed in Section 2, these items correspond to the information generally available in decentralized recommendation systems [11], but may also be obtained in a centralized setting if the associated website publishes user ratings. We observe that this is a fairly strong assumption on the attacker as in many cases, available information is much less precise. For example, social network plugins may publish updates such as "Tom just saw Highlander", but without specifying a rating.

Even though the auxiliary knowledge comprises both items and ratings, we define the attack's goal as the need to determine whether the target profile contains a particular item, with a high-enough rating (≥ 3).

We evaluate the attack according to two families of metrics. The first one measures the ability of Sybils to construct the ideal neighborhood to carry out the attack. As discussed, this consists of the target user plus $k-1$ Sybils. Specifically, we measure (i) the fraction of Sybils that obtain such an ideal neighborhood, and (ii) the fraction of Sybils whose neighborhoods contain the target.

The second family of metrics measures instead the success of the attack itself and consists of *yield* and *accuracy*. Yield simply measures the number of guesses that the attacker can make thanks to the action of the Sybils. Accuracy measures instead the fraction of correct guesses.

4. RESULTS

The attack aims at populating the neighborhoods of each of the Sybil identities with the target user and exactly $k-1$ other Sybils. Although this may seem straightforward,

the ability to do so depends on the similarity of the target with other real users. Our analysis of the ML-100k dataset highlighted that a very large proportion of users have perfect or almost perfect homologous users that effectively protect them from the action of Sybils.

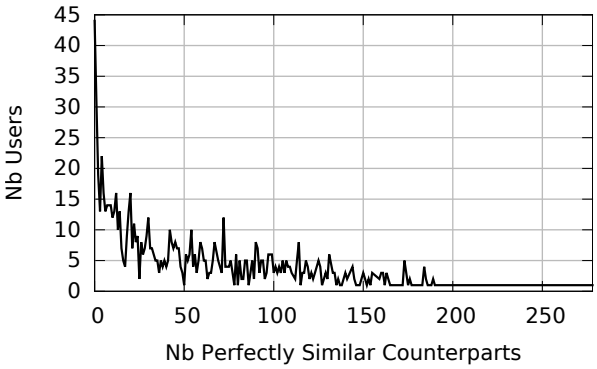

Figure 1: Distribution of the number of perfectly similar counterparts for the users in ML-100k.

As described in Section 3, Mahout, like many other recommender systems, uses a variant of cosine similarity that only considers the ratings for items that appear in both user profiles. This means that two users may have a perfect similarity (i.e. a similarity of 1) even if their item sets are not exactly identical. It is enough for them to have expressed the same ratings on their common items. Figure 1 visualizes this observation by plotting the distribution of the number of users that have perfectly similar profiles in the dataset, or perfectly similar counterparts. The point at $(0, 45)$ means that only 45 out of 943 users in the dataset are not perfectly similar to any other user. In the following, we analyze how this user distribution affects the performance of the Sybil attack in two scenarios: an attacker targeting a single user, and multiple parallel attacks targeting a percentage of users.

4.1 Isolated Attack

We start our analysis by evaluating the effectiveness of an isolated Sybil attack. We consider a neighborhood size of $k = 10$, and a "team" of 10 Sybils, each aiming to have a neighborhood consisting of the target plus the 9 other Sybils.

As expected, the presence of so many mutually similar users severely limits the performance of the attack. Figure 2 plots the fraction of Sybil identities that obtain exactly the neighborhood that they were expecting to obtain, as a function of the number of perfectly similar counterparts of their target user. Each line corresponds to a different amount of auxiliary knowledge made available to the attacker, expressed as a percentage of the target's profile. The plot shows that none of the Sybils obtains its ideal neighborhood when attacking users that have at least 2 perfectly similar counterparts, regardless of the amount of auxiliary knowledge they have. Moreover, even when attacking users that have no perfectly similar counterparts, the percentage of Sybils that get an ideal neighborhood strongly depends on the amount of available auxiliary knowledge. If a Sybil knows less than 50% of the ratings in the target profile, then it basically has no chance to obtain its ideal neighborhood.

While these results seem catastrophic for the Sybil attack, Sybils do not need to have perfect neighborhoods to extract knowledge about their target. A Sybil that only has the target and other Sybils in its neighborhood can be sure that every recommendation it receives correspond to an item from the target profile. But even if its neighborhood contains unwanted users, the Sybil can still make reasonable guesses provided that its neighborhood contains the target. For this reason, Figure 3 complements the above results by presenting the fraction of Sybil neighborhoods that contain the target user. The plot shows that even with 30% of auxiliary knowledge, approximately 25% of the Sybils manage to have the target in their imperfect neighborhoods. This still allows them to identify potential matches for the target's interests even if with lower accuracy.

Figures 4 and 5 conclude our analysis of the isolated attack by presenting the yield and accuracy of the predictions made by the Sybils. To measure these, we have each Sybil request 5 recommendations from its neighborhood. This gives a maximum possible yield of 50—5 recommendations for each of the 10 Sybils. Figure 4 shows that the attack's yield decreases as the number of auxiliary items increases. This is not surprising, the more the attacker knows about the target, the fewer recommendations the Sybils can get. Moreover, for high percentages of auxiliary knowledge, the yield value increases with the number of perfectly similar counterparts of the target. This results from the presence of non-target users in the Sybil's neighborhoods. This presence causes the Sybils to receive more recommendations, albeit not necessarily accurate ones as shown in Figure 5.

As expected, accuracy decreases with the number of perfectly similar counterparts of the target. Moreover it is also roughly proportional to the amount of auxiliary knowledge available to the attacker. In any case, when auxiliary items cover 50% of the target's profile, the attacker only guesses the target's missing items with less than 50% of accuracy.

These results highlight a sharp contrast with those of [13] and [28]. Both papers suggest that, in sparse datasets, a set of auxiliary items of about $log(N)$, N being the number of users, should be enough to identify the target. As highlighted by Figure 1 and the following, this is clearly not the case in MovieLens-100k.

4.2 Parallel Attacks Against Multiple Users

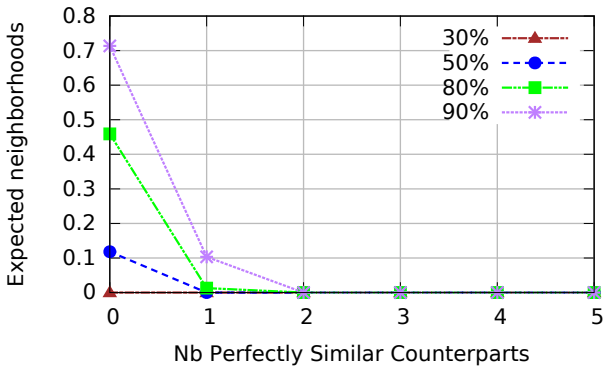

Figure 2: Fraction of Sybils that obtain their ideal neighborhoods when performing an isolated attack, for different percentages of auxiliary items.

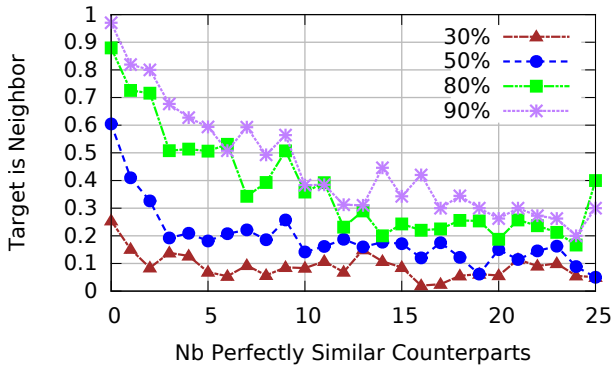

Figure 3: Fraction of Sybils whose neighborhoods contain the target user when performing an isolated attack for various percentages of auxiliary items.

Next, we evaluate how the performance of the attack varies in the presence of multiple sets of Sybils targeting multiple users. Figure 6 depicts the fraction of Sybils that manage to obtain a perfect neighborhood as a function of the fraction of auxiliary items with varying numbers of target user: 1 user as in Section 4.1, as well as 25%, 50%, and 100% of the user. The values in Figure 6 and in all the subsequent figures refer to the target users without perfectly similar counterparts. When attacking multiple users, the attacker deploys a separate team of $k = 10$ Sybils for each target.

The plot shows that the fraction of perfect neighborhoods decreases with the number of simultaneously attacked targets. By attacking multiple users that possibly overlap in terms of similarity, different teams of Sybils may interfere with each other causing neighborhoods to contain Sybils from other teams instead of the target user.

Figure 7 confirms this reasoning and shows that the fraction of neighborhoods containing the target drastically drops when moving from 1 to 25% of the user as targets. When attacking 50% of the user, only a few percents of the Sybils have the respective targets in their neighborhoods.

Figures 8 and 9 confirm the negative impact of parallel attacks. Accuracy drops significantly as the number of concurrent attacks increases.

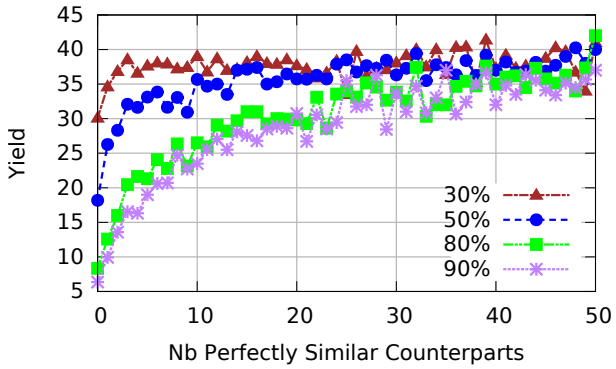

Figure 4: Yield obtained by the isolated attack for various percentages of auxiliary items.

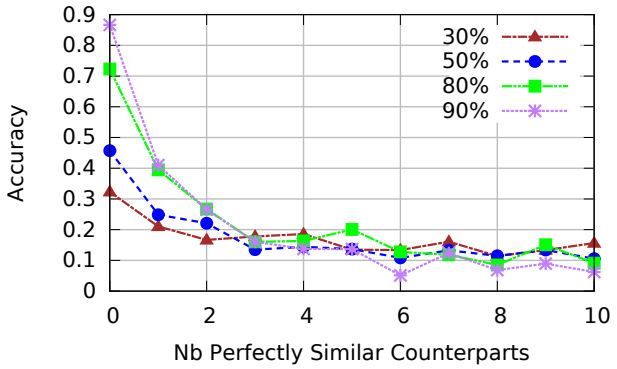

Figure 5: Accuracy obtained by the isolated attack for various percentages of auxiliary items.

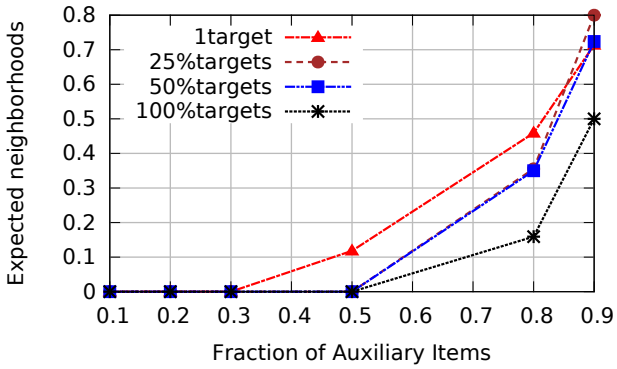

Figure 6: Fraction of Sybils with perfect neighborhoods with a variable number of targets.

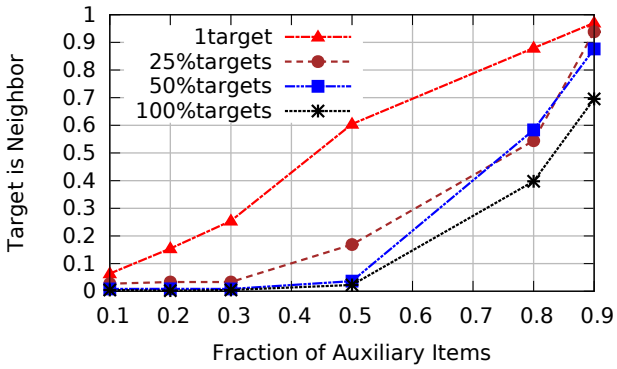

Figure 7: Fraction of Sybils that have the target in their neighborhoods with a variable number of targets.

5. RELATED WORK

Several authors have recognized the trade-off between accuracy and privacy in recommenders [26, 30, 22]. This has led to results along two main lines of research: identifying potential threats and attacks, and improving recommenders to make them more resilient to such attacks.

5.1 Attacks against Recommenders

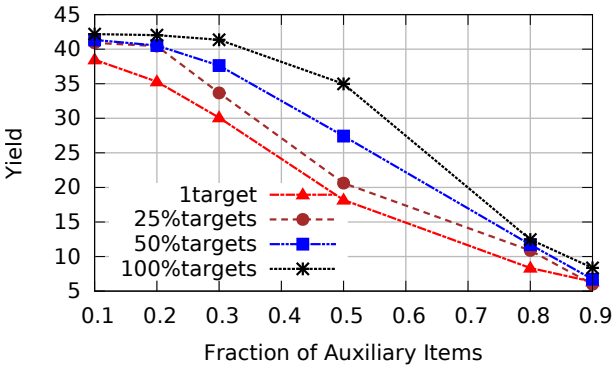

Figure 8: Yield with a variable number of targets.

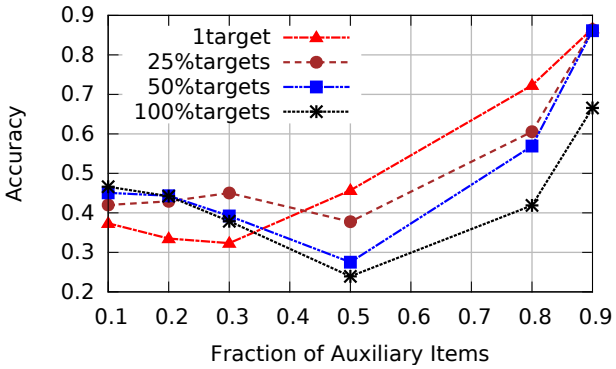

Figure 9: Accuracy with a variable number of targets.

Within the context of attacks, we can distinguish passive and active ones. In the former, the attacker operates as a normal user and simply tries to learn information about other users through legitimate means. In the latter, the attacker carries out operations that go outside the standard behavior of a user for example by introducing fake identities like in the attack of this paper, or fake items.

Within the domain of passive attacks, [13] analyzes how auxiliary information, obtained from the system itself or from external sources, makes it possible to extract individual user preferences from otherwise aggregate information such as related-item lists or item-covariance matrices. The authors also introduce the active attack we consider in this paper, but they do not analyze or evaluate its effectiveness.

BlurMe [33] and [9] present passive attacks that extract demographic information such as the ethnicity or gender of users from the ratings of items in a recommender. BlurMe also proposes an obfuscation mechanism to limit the impact of such an attack, while [9] also develops an active attack that maximizes the ability to learn new information by asking users to rate specific items. Pistis [24] considers an active attack similar to the one described in this paper, and proposes a mechanism that limits its impact by expressing ratings on privacy-preserving groups of items.

Shilling attacks [18, 32] also adopt the active model but with a different purpose: biasing the output of the recommender, for example to favor the products of a particular brand. In the context of user-based systems, [18] shows that even so-called privacy-preserving collaborative-filtering systems remain vulnerable to profile injection (shilling) attacks. In the item-based context, [32] shows that the injection of specific item profiles (power items) can equally bias the output of the recommender.

5.2 Privacy-Preserving Recommenders

The first attempts to provide privacy-preserving recommenders have focused on decentralized solutions. One of the first papers, [15] exploits homomorphic encryption in a peer-to-peer environment to compute similarities securely while eliminating the need for a Big Brother. Other authors, including some authors of this paper, have instead proposed anonymization [12], and profile-obfuscation [8] techniques. In a centralized setting, some authors [6, 29] have proposed injecting noise into user profiles to build privacy-preserving data-mining and recommendation algorithms. But these schemes have turned out to be vulnerable to statistical attacks that filter out the random noise to reconstruct the missing information [5, 21, 23]. Moreover, all the above solutions concentrate on hiding user profiles. They therefore remain vulnerable to attacks that combine recommended items with auxiliary information available through external sources, like the one we study in this paper.

Systems that apply differential privacy only to neighborhood computation [34] exhibit the same problem. But some authors have also proposed systems that incorporate randomization and ensure differential privacy when they generate recommendations [10, 27]. In [10], some of the authors of this paper demonstrate that their approach can effectively counteract a Sybil-based censorship attack. However, its effectiveness against an attacker equipped with external auxiliary information remains unclear.

6. CONCLUDING REMARKS

We analyzed the impact of an active Sybil attack on user-based collaborative-filtering recommenders. Our results show that while the attack is generally effective, some users receive natural protection from their inherent similarity with other users in the system. Our analysis of the MovieLens-100k dataset shows that a large proportion of users in the KNN graph are naturally surrounded by other users that protect them from the action of Sybil identities. This is in sharp contrast with what is highlighted in [13], namely that $log(N)$ auxiliary items suffice to identify the target.

Despite our interesting results, this work remains a preliminary effort. We plan to extend our analysis in several ways. First, we aim to confirm our current results on different datasets and on different variants of the recommender. For example, we used a version of cosine similarity that only accounts for items that appear in both the profiles being considered. Counting items that appear only in one of the profiles might have a negative impact on our results.

Second, we aim at exploring variants of the attack protocol. For example, we will study the case of adaptive Sybils. The effect of users' inherent similarity is probably even more important in the evolution of adaptive Sybil neighborhoods. Finally, we have started varying the nature of the information available to the attacker. In many cases, an attacker won't have access to the full ratings but will have access to the set of items. Our initial tests indicate that this less precise information makes the attack even less effective than what we show in this paper.

7. ACKNOWLEDGEMENTS

This work was partially funded by the Region of Brittany, France, by the French ANR project SocioPlug (ANR-13-INFR-0003), by the DeSceNt project granted by the Labex CominLabs excellence laboratory (ANR-10-LABX-07-01), and by the Google Focused Research Award Web Alter-Ego.

8. REFERENCES

[1] Facebook. `https://www.facebook.com/`.

[2] Last.fm. `https://www.last.fm/`.

[3] Linkedin. `https://www.linkedin.com/`.

[4] Mahout. `https://mahout.apache.org/`.

[5] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *PODS*. ACM, 2001.

[6] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *SIGMOD*. ACM, 2000.

[7] J. Bennett and S. Lanning. The netflix prize. In *SIGKDD*. ACM, 2007.

[8] S. Berkovsky, Y. Eytani, T. Kuflik, and F. Ricci. Enhancing privacy and preserving accuracy of a distributed collaborative filtering. In *RecSys*. ACM, 2007.

[9] S. Bhagat, U. Weinsberg, S. Ioannidis, and N. Taft. Recommending with an agenda: Active learning of private attributes using matrix factorization. In *RecSys*. ACM, 2014.

[10] A. Boutet, D. Frey, R. Guerraoui, A. Jégou, and A. Kermarrec. Privacy-preserving distributed collaborative filtering. In G. Noubir and M. Raynal, editors, *NETYS*, volume 8593 of *Lecture Notes in Computer Science*. Springer, 2014.

[11] A. Boutet, D. Frey, R. Guerraoui, A. Jégou, and A.-M. Kermarrec. WHATSUP: A decentralized instant news recommender. In *IPDPS*. IEEE, 2013.

[12] A. Boutet, D. Frey, A. Jégou, A.-M. Kermarrec, and H. B. Ribeiro. Freerec: An anonymous and distributed personalization architecture. In *NETYS*. Springer, 2013.

[13] J. Calandrino, A. Kilzer, A. Narayanan, E. Felten, and V. Shmatikov. "you might also like:" privacy risks of collaborative filtering. In *SP*. IEEE, 2011.

[14] J. Canny. Collaborative filtering with privacy. In *SP*. IEEE, 2002.

[15] J. Canny. Collaborative filtering with privacy via factor analysis. In *SIGIR*, 2002.

[16] A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. In *WWW*. ACM, 2007.

[17] D. Goldberg, D. A. Nichols, B. M. Oki, and D. B. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[18] I. Gunes, A. Bilge, and H. Polat. Shilling attacks against memory-based privacy-preserving recommendation algorithms. *TIIS*, 7(5):1272–1290, 2013.

[19] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR*. ACM, 1999.

[20] T. R. Hoens, M. Blanton, and N. V. Chawla. Reliable medical recommendation systems with patient. *IHI*, 2010.

[21] Z. Huang, W. Du, and B. Chen. Deriving private information from randomized data. In *SIGMOD*. ACM, 2005.

[22] A. Jeckmans, M. Beye, Z. Erkin, P. Hartel, R. Lagendijk, and Q. Tang. Privacy in recommender systems. In *Social Media Retrieval*, Computer Communications and Networks, pages 263–281. Springer, 2013.

[23] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *ICDM*. IEEE, 2003.

[24] D. Li, Q. Lv, H. Xia, L. Shang, T. Lu, and N. Gu. Pistis: A privacy-preserving content recommender system for online social communities. In *WI-IAT*. IEEE, 2011.

[25] G. Linden, J. Jacobi, and E. Benson. Collaborative recommendations using item-to-item similarity mappings, July 24 2001. US Patent 6,266,649.

[26] A. Machanavajjhala, A. Korolova, and A. D. Sarma. Personalized social recommendations: accurate or private. *VLDB*, 2011.

[27] F. McSherry and I. Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *SIGKDD*. ACM, 2009.

[28] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *SP*. IEEE, 2008.

[29] H. Polat and W. Du. Svd-based collaborative filtering with privacy. In *SAC*. ACM, 2005.

[30] N. Ramakrishnan, B. J. Keller, B. J. Mirza, A. Y. Grama, and G. Karypis. Privacy risks in recommender systems. *IEEE Internet Computing*, 5(6):54–62, Nov. 2001.

[31] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *CSCW*. ACM, 1994.

[32] C. E. Seminario and D. C. Wilson. Attacking item-based recommender systems with power items. In *RecSys*. ACM, 2014.

[33] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft. BlurMe: Inferring and obfuscating user gender based on ratings. In *RecSys*. ACM, 2012.

[34] T. Zhu, G. Li, Y. Ren, W. Zhou, and P. Xiong. Differential privacy for neighborhood-based collaborative filtering. In *ASONAM*. ACM, 2013.