Privacy, Shilling, and The Value of Information in Recommender Systems

Shyong K Lam and John Riedl

GroupLens Research
Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
{lam,riedl}@cs.umn.edu

Abstract. Recommender systems are an increasingly popular tool used by many consumers to help deal with information overload in today's marketplace. At the cost of some personal information, these systems are able to personalize a user's online experience and guide them toward making better decisions. This paper examines two issues relating to privacy in recommender systems: the value of information and shilling. The paper considers the privacy cost of recommender systems and proposes ways that the loss of privacy can be limited and balanced against recommendation and personalization quality.

1 Introduction

Consumers in today's marketplace are often overwhelmed with the number of different options available to them. To combat this problem, which is often called *information overload*, many people have turned to recommender systems. These are tools that use opinions about items in some information domain in order to make recommendations to a user regarding which items she may wish to consider. One example of a recommender system is *MovieLens* (http://www.movielens.org), which is a system that makes personalized recommendations suggesting movies that a user might like based on the movies she has seen and has expressed an opinion about. Using recommender systems, online retailers can personalize their virtual storefronts specifically for each consumer to help them find and buy things.

Unfortunately, while recommenders provide benefit to users as a decision-making tool, this benefit often comes with a cost in privacy. Personalized recommenders like *MovieLens* require some information about the user's taste before they are able to make personalized recommendations. (Some recommenders, such as *Amazon.com*'s customer reviews, provide recommendations with no information about the user's tastes, but such systems are not truly personalized — everybody sees the same reviews.)

In our work, we largely focus on personalized recommender systems. In particular, we are interested in ones that use automated collaborative filtering (ACF), which refers to algorithms that generate recommendations on the

basis that people who have expressed similar opinions in the past are likely to share opinions in the future. Two commonly used ACF algorithms are user-based k-Nearest-Neighbor[1] and item-based k-Nearest-Neighbor[2]. These algorithms are commonly classified as *memory-based* algorithms where the entire set of preference information is used in producing recommendations.

A second class of ACF algorithms consists of *model-based* algorithms in which the preference information is processed into a model that can in turn be used to compute recommendations. An example of such an algorithm is the one based on Bayesian networks as described by Breese et al. [3]. These models contain the preference information in a reduced form, and it is difficult – and perhaps impossible – to recover the original preference information given just the model. Using a model-based algorithm allows the recommender system to preserve its users' privacy in that it can distribute the models to its users who can in turn compute recommendations without directly providing preference information to the server. Model-based algorithms of this type make possible the separation of data about *self* from data about *others*. In principle, data about self might be maintained on the client, and not shared with the centralized server. However, most model-building algorithms retain the requirement that sufficiently many users must provide the system with preference information in order to have a high-quality model built in the first place.

There has been substantial theoretical work done in studying this and other privacy issues in ACF-based recommender systems. Ramakrishnan et al [4] describe a graph-theoretic model showing how information can be inferred about certain classes of users by observing the recommendations made by a system. Canny [5] and Miller et al. [6] suggest new privacy-preserving paradigms for recommender systems that use peer-to-peer and encryption to prevent the system operator from learning any explicit preference information about its users, and to limit what users can learn about each other. Even with these restrictions on information flow, the proposed systems are able to produce recommendations with accuracy comparable to existing non-privacy-preserving systems.

We seek to extend this work by exploring steps to promote privacy that can be taken with currently-deployed systems that may have already collected a significant amount of data about their users. Most of the discussion applies equally to memory-based or model-based algorithms. The theme of this paper is to informally explore the costs and benefits of limiting the amount of information kept by a recommender about a user. By using information-theoretic techniques, we believe it may be possible to selectively decide which information about a user to retain and which to discard while being able to perform a reasonable amount of personalization. In doing this, however, it is possible that the recommender system becomes more vulnerable to malicious attacks by third parties.

2 Value of Information

A personalized recommendation algorithm typically requires input from the user population in order to make recommendations. In general, the more information

that is known about the users, the more accurate the recommendations are. This presents a conundrum when considering privacy in systems that use such algorithms — more data leads to better personalization, but acquiring data can be invasive to user privacy. Ideally, one would like to find a balance where the system is able to make good recommendations while not requiring users to give up too much information about themselves.

The accuracy of an algorithm with respect to the amount of information known about the user follows a diminishing returns curve. That is, once a certain amount is known about a user, obtaining further information is only marginally useful. This raises the possibility of finding a "sweet spot" that maximizes the recommendation accuracy per unit of information known about the user.

Some people, particularly advertisers, seek to provide personalization based on a small amount of information. For instance, recommendations might be based on demographic data (e.g. the usual "ZAG" information — zip code, age, gender), or generalized preferences of attributes describing the items involved (in movies, this might mean the user's favorite genres). These personalization methods only have a modest effect on privacy as the information given up by the user cannot easily reveal the user's identity.

In contrast, highly personalized recommenders, such as those based on ACF, require a high degree of personal preference information from the user. These requirements lead to larger privacy concerns since this level of preference information may reveal substantial personal information about the user. There are many open privacy-related research questions around the elicitation of such preference information, including: (a) How much information is given up by a user when she provides an opinion about an item? (b) Does the amount of information content vary by item? (c) How does the privacy lost relate to the information gained by the recommender system?

Note that the answers to these questions are likely to vary by domain. In some domains, such as music CDs, users may be relatively open to sharing their tastes with others. In other domains, such as medical information, users may have serious concerns about sharing their preferences with anyone, because of the potential harm should the information leak to colleagues or potential employers. In still other domains, such as scientific research papers, the sensitivity of the information may vary with time. While working on a paper, a researcher may not want others to know what related work she is studying; once the paper is published, the list of references is publicly available and no longer presents a privacy concern. In the discussion we try to focus on aspects of privacy in recommenders that are likely to be domain-independent.

Intuitively, the goal of preference information is to differentiate a user from her peers. Preference information that does a better job of differentiating among users should be inherently more useful in personalizing a system for that user. For instance, knowing that a user likes the movie "Toy Story" reveals less about her than knowing that she likes "Fahrenheit 9/11." The former is a universally-liked movie, while the latter is more polarizing with a higher level of disparity among users' opinions. This is the basis of an idea proposed by Pennock and

Horvitz that says if one can calculate how useful a given piece of information is, that is, if one had a "value-of-information" metric, then one can tune a system to optimize its data collection process by choosing to solicit user preferences on items that carry the most value [7].

In past work we have explored the related issue of eliciting information from new users in a recommender system in ways that optimize both the required user effort and initial recommendation accuracy [8, 9]. Using the value of information (VOI) concept, we were able to build VOI-aware MovieLens interfaces that reduced the user effort needed to start receiving recommendations. Moreover, the accuracy of those recommendations was improved compared to ones made based on the initial user models built for new users that did not use the VOI-aware interface.

In the interest of user privacy, this kind of approach may be comforting to some users in that fewer discrete pieces of information (e.g. movie ratings) need to be provided before the system becomes accurate. However, the information theory involved says that the user has given up an equivalent amount of information about herself as she would have with an unoptimized approach. On the other hand, it is possible to use VOI to measure how much information is needed to make good recommendations, and then to stop collecting new information from the user once that point is reached. More generally, a recommender system can use VOI to bound the amount of information collected about a user to some optimal level with respect to both privacy and recommendation quality.

These ideas lead to the following set of questions about the role of VOI in preserving user privacy for existing users in a recommender system that may comprise an interesting follow-up program of research:

2.1 Amount of Data

How much data is needed from a user to make good recommendations? How do we calculate the "sweet spot"? It is desirable for the system to know this so that it could stop eliciting data from a user when it feels it has built a sufficiently good user model. A related issue here is how this may change over time. As the system or the user evolves, or as new items are introduced, will more information be needed to maintain high-quality recommendations?

One challenge with limiting the amount of data provided by a user is that the system might then recommend items the user already possesses. For instance, a recommender for music CDs might prefer to filter out from recommendation lists CDs the user already owns. To support this feature, the system might encourage the user to upload complete information about her collection. The user would then have to balance her preference for privacy with her preference for personally filtered recommendations. One privacy-preserving solution might be to have a client-side filter select from among the recommended items those to display to the user. The centralized system could then form recommendations based on the modest amount of data it needs for quality recommendations.

2.2 User Interface

What should the user interface look like, especially after the system thinks it has learned enough about the user? What if the user *wants* to tell us more about herself? How does one present a privacy-preserving recommender system in an understandable way? In our experiences with MovieLens, we have found no shortage of people willing to provide hundreds and sometimes thousands of movie ratings. Indeed, user feedback from our periodic surveys reveals that rating movies is among the leading reasons people have for using the system! These observations that some users are perfectly willing to give up their privacy may make it tricky to create a usable interface that effectively conveys the privacy-preserving aspects of the recommender system.

2.3 Selectively Discarding Data

If we find ourselves knowing "too much" about a user, which particular pieces of data should be kept? This is strongly tied to VOI, but the answer to this question is not necessarily the information with the highest value. If "low-valued" information is discarded from many users' models, then perhaps that information is no longer low-valued since it has become more rare. Choosing an appropriate set of data that balances both the benefit to the overall system and the quality of each individual user model seems to be a challenging task.

2.4 Impact on CF Algorithms

How well do current collaborative filtering algorithms operate in reduced-data environments? Clearly, both recommendation accuracy and coverage might be affected. Some algorithms such as SVD seem more naturally suited for sparse data sets [10] — are they even better if given selectively chosen data? Is the sparse data inherently less noisy, and if so, could it even lead to *better* recommendations using specialized algorithms? Another effect might be that the algorithm is more susceptible to shilling attacks. This will be described in further detail in section 3.

3 Shilling

One of the primary uses for a recommender system is to help people make decisions, whether it be which movie to see, which restaurant to eat at, or what web site to visit. Naturally, this makes recommender systems very interesting to people with vested interests in what people choose to consume. For instance, a restaurant owner would be more successful if more people ate at his establishment, so it is within his best interests to have it recommended often by a restaurant recommender system. One way to do this is to provide good service to garner a good reputation among restaurant diners. In turn, this would likely lead to more frequent recommendation as users express high opinions of the restaurant on the recommender system.

However, a more underhanded and often cheaper way to increase recommendation frequency is to manipulate or trick the system into doing so. This can be done by having a group of users (human or agent) use the recommender system and provide specially crafted "opinions" that cause it to make the desired recommendation more often. Instances of these manipulations have been observed in the past. For example, it has been shown that a number of book reviews published on Amazon.com are actually written by the author of the book being reviewed[1]. A consumer trying to decide which book to purchase could be misled by such reviews into believing that the book is better than it really is. The privacy Amazon provides to its reviewers allows this to happen, as reviews are usually only attributed to a pseudonym. Unfortunately, there is no clear-cut solution here since reviewers often wish to maintain their privacy, which can conflict with the reader's desire to be able to know how much they can believe a review. As a step in solving this particular problem, Amazon has introduced a "Real Name" feature to their review system that specially identifies reviewers who have elected to divulge their "true" identity, and thus might be more trustworthy.

Our previous work [11] has shown that these types of "shilling" attacks are indeed effective and are relatively easy to carry out with collaborative filtering algorithms in use today. An attacker only needs to know a small amount of information about the user and item population in order to perform an attack that increases the number of times some particular item (or set of items) is recommended. Furthermore, the attacks are non-trivial to detect with typical measures of recommender system performance.

A concern of introducing more privacy to a recommender system is whether it might make these attacks easier or more common, as the pseudonymity or anonymity provided by privacy usually invites abuse of this nature (consider what usually happens on any popular web-based forum or bulletin board when no controls are placed on posting messages). In particular, the issue of privacy in recommender systems raises the following questions and concerns regarding shilling attacks.

3.1 Attack Effectiveness

Do shilling attacks become more effective against privacy-preserving recommender systems? As additional privacy is introduced to a recommender system, the opportunities for attacks can increase considerably. Our work [11] shows that attacks that target recommendation frequency of low-information items (i.e. ones with few ratings) are more effective than attacks against high-information items. In a system that tries to maintain a minimal amount of information about its members, it is possible that *every* item might have sufficiently few ratings to be vulnerable to highly-effective attacks.

[1] http://www.onlinesecurity.com/links/links837.php

3.2 Attack Difficulty

Are shilling attacks more or less difficult to mount against privacy-preserving recommender systems? As mentioned above, more individual items might become ideal targets for effective attacks. On the other hand, if the recommender system only keeps some subset of data provided by each user, an attack strategy will need to take that into consideration, both for the users being targetted and for the "false" users introduced by the attack. This would likely require the attacker to know more about the system being attacked, thus increasing the cost of an attack.

Another possible impeding factor in an attack is the interface presented to users. A VOI-aware interface such as the ones used in our past work [8, 9] can control which items may be rated by a user in order to maximize the information gain per collected rating. This significantly constrains what an attacker can do and could make it more difficult to impact the system in precise ways.

3.3 Attack Detection

In a privacy-preserving recommender system, is it easier or harder to detect an attack? One might theorize that in a low-data environment, it becomes easier to identify atypical patterns that are indicative of an attack. If found to be true, this would certainly be a boon to recommender system operators. On the other hand, discarding some of the data entered by a shilling agent might leave the remaining data looking "more human," and hence harder to detect.

4 Conclusion

The issue of privacy in recommender systems is a rich area with many aspects that have yet to be explored. Recommenders – especially highly personalized recommenders like those using automated collaborative filtering – raise important issues about how the data they collect impinges on user privacy.

In this paper we explored two aspects of recommender systems that relate to these important privacy questions: VOI and shilling. Previous work on VOI shows that it can be used to more effectively collect information from new users. We believe it can similarly be used to determine when to stop collecting information to properly balance the privacy given up by users with the quality of the recommendations, and to intelligently choose which information to discard if "too much" is known about a user. The challenge of shilling is that the aforementioned privacy protections may make shilling easier, especially if they reduce the amount of information the recommender system keeps about each user.

This is, however, just the tip of the iceberg. There are many other aspects of privacy in recommender systems, including its role in community-oriented systems where participants interact with each other on a regular basis, or in small-world systems in which most or all of the participants know each other. Here, privacy might not be as big a concern to users as it might be with, say,

a large corporate-owned system. In fact, users might even want ways to share validated personal information, so they know something about who they are interacting with. Successful systems will likely have layers of privacy, with users in charge of what they see from others and what they show to others.

5 Acknowledgments

Thanks to the anonymous workshop referees for their helpful suggestions, and to members of GroupLens Research at the University of Minnesota for many fruitful discussions. Particular thanks are due to our colleagues Al Mamunur Rashid, Istvan Albert, Dan Cosley, Sean M. McNee, and Joseph Konstan, our co-authors on the VOI research [8, 9]. This work was supported by grants from the NSF (DGE 95-54517, IIS 96-13960, IIS 97-34442, IIS 99-78717, and IIS 01-02229).

References

1. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work, Chapel Hill, North Carolina, United States, ACM Press (1994) 175–186
2. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: WWW '01: Proceedings of the tenth international conference on World Wide Web, Hong Kong, Hong Kong, ACM Press (2001) 285–295
3. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98). (1998) 43–52
4. Ramakrishnan, N., Keller, B.J., Mirza, B.J., Grama, A., Karypis, G.: Privacy risks in recommender systems. IEEE Internet Computing **5** (2001) 54–62
5. Canny, J.: Collaborative filtering with privacy via factor analysis. In: SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, Tampere, Finland, ACM Press (2002) 238–245
6. Miller, B.N., Konstan, J.A., Riedl, J.: Pocketlens: Toward a personal recommender system. ACM Transactions on Information Systems **22** (2004) 437–476
7. Pennock, D.M., Horvitz, E., Lawrence, S., Giles, C.L.: Collaborative filtering by personality diagnosis: A hybrid memory and model-based approach. In: UAI '00: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, Stanford, CA, Morgan Kaufmann Publishers Inc. (2000) 473–480
8. Rashid, A.M., Albert, I., Cosley, D., Lam, S.K., McNee, S., Konstan, J.A., Riedl, J.: Getting to know you: Learning new user preferences in recommender systems. In: Proceedings of the 2002 International Conference on Intelligent User Interfaces, San Francisco, CA (2002) 127–134
9. McNee, S.M., Lam, S.K., Konstan, J.A., Riedl, J.: Interfaces for eliciting new user preferences in recommender systems. In: User Modeling, Johnstown, PA, USA, Springer Verlag (2003) 178–187

10. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Application of dimensionality reduction in recommender system – a case study. In: ACM WebKDD 2000 Web Mining for E-Commerce Workshop. (2000)
11. Lam, S.K., Riedl, J.: Shilling recommender systems for fun and profit. In: WWW '04: Proceedings of the 13th international conference on World Wide Web, New York, NY, USA, ACM Press (2004) 393–402