Segment-Based Injection Attacks against Collaborative Filtering Recommender Systems*

Robin Burke, Bamshad Mobasher, Runa Bhaumik, Chad Williams
Center for Web Intelligence, DePaul University
School of Computer Science, Telecommunication, and Information Systems
Chicago, Illinois, USA
{rburke, mobasher, rbhaumik, cwilli43}@cs.depaul.edu

Abstract

Significant vulnerabilities have recently been identified in collaborative filtering recommender systems. Researchers have shown that attackers can manipulate a system's recommendations by injecting biased profiles into it. In this paper, we examine attacks that concentrate on a targeted set of users with similar tastes, biasing the system's responses to these users. We show that such attacks are both pragmatically reasonable and also highly effective against both user-based and item-based algorithms. As a result, an attacker can mount such a "segmented" attack with little knowledge of the specific system being targeted and with strong likelihood of success.

1. Introduction

Recent research has begun to examine the vulnerabilities and robustness of different recommendation techniques, such as collaborative filtering, in the face of what has been termed "shilling" attacks [2, 1, 5, 6], but we call *profile injection attacks*, since promoting a particular product is only one way such attack might be used. In a profile injection attack, an attacker interacts with the recommender system to build within it a number of profiles associated with fictitious identities with the aim of biasing the system's output.

It is easy to see why collaborative filtering is vulnerable to profile injection attacks. A user-based collaborative filtering algorithm collects user profiles, which are assumed to represent the preferences of many different individuals and makes recommendations by finding peers with like profiles. If the profile database contains biased data (many profiles all of which rate a certain item highly, for example), these biased profiles may be considered peers for genuine users and result in biased recommendations. This is precisely the effect found in [5] and [6].

Researchers who have examined this phenomenon have concentrated on broad attack models whose profiles contains ratings across the spectrum of available objects and have measured their results by looking at how all of the users of the system are affected in the aggregate. However, it is a basic truism of marketing that the best way to increase the impact of a promotional activity is to target one's effort to those already predisposed toward one's product. In other words, it is more likely that an attacker wishing to promote a particular product will be interested not in how often it is recommended to all users, but how often it is recommended to the particular market segment that is likely to already have a propensity to purchase it.

This paper examines a particular attack model that we call the *segmented attack* in which the attacker concentrates on a set of items of similar content that have high visibility, the *Harry Potter* series being a good example in the book domain. It is certainly the case that these books are highly popular and widely read – it would follow that they would be rated by many users of a collaborative system. Users who enjoy these books are likely to share some characteristics: they may be children or parents who have an interest in exciting fantasy stories involving magic. These facts are general knowledge about the book domain readily available outside of any particular recommender system, which means that the degree of system-specific knowledge required to mount the attack is relatively low. We show that the segmented attack is both effective and practical against both user-based and item-based collaborative filtering algorithms.

*This research was supported in part by the National Science Foundation Cyber Trust program under Grant IIS-0430303.

2. The Segmented Attack

A profile injection attack against a collaborative recommender system consists of a set of attack profiles, biased profile data associated with fictitious user identities, and a target item, the item that the attacker wishes the system to recommend more highly (a *push* attack), or wishes to prevent the system from recommending (a *nuke* attack). We concentrate on push attacks in this paper. An attack model is an approach to constructing the attack profile, based on knowledge about the recommender system, its rating database, its products, and/or its users.

Prior work on recommender system stability has examined primarily three attacks. The sampling attack from [6] is primarily of theoretical interest as it requires the attacker to have access to the ratings database itself. The random attack [5] forms profiles by associating a positive rating for the target item with random values for the other items. The average attack [5] assumes that the attacker knows the average rating for each item in the database and assigns values randomly distributed around this average, except for the target item. This attack has been found to be effective against user-based collaborative recommendation algorithms, but less so against item-based recommendation.

Each of these prior attack models assumes that the attacker is interested in associating the pushed item with any profile in the database. This makes for a simple attack model. However, suppose the attacker Eve has written a fantasy book for children. She would no doubt prefer that her book be recommended to buyers who had expressed an interest in this genre, for example buyers of *Harry Potter* books, rather than buyers of books on Java programming or motorcycle repair. Eve would rightly expect that the "fantasy book buyer" segment of the market would be more likely to respond to a recommendation for her book than others.

To target the users in the segment, the attacker constructs profiles with high ratings for items that are preferred by users in the targeted market segment and low ratings for other items. These profiles will tend to match the in-segment users who also have a strong preference for these items. An attacker like Eve only needs to know what books are both similar to the one she wants to push and relatively popular in order to generate such profiles.

3. Recommendation Algorithms

This paper reports on results for two of the most commonly-used collaborative algorithms: user-based and item-based collaborative recommendation using nearest-neighbor techniques.[3, 7] The standard collaborative filtering algorithm is based on user-to-user similarity [3]. This kNN algorithm operates by selecting the k most similar users to the target user, and formulates a prediction by combining the preferences of these users. Similarity is measured using Pearson's r-correlation coefficient: similar users are those whose profiles are highly correlated with each other. In our implementation, we have used a value of 20 for the neighborhood size, and we filter out all neighbors with a similarity of less than 0.1.

Item-based collaborative filtering works by comparing items based on their pattern of ratings across users. Again, a nearest-neighbor approach can be used, with the most common similarity metric the adjusted cosine similarity measure introduced by [7]. In this measure, all user profiles are normalized by subtracting the user's mean rating. When items are compared, the ratings given by each user to that item are combined in a vector and the similarity between them is calculated as the vector cosine. After computing the similarity between items we select a set of k most similar items to the target item and generate a predicted value, weighting the user's known rating for each similar item by its similarity value. We consider a neighborhood of size 20 and ignore items with negative similarity.

4. Experiments

In our experiments we have used the publicly-available Movie-Lens 100K dataset[1]. This dataset consists of 100,000 ratings on 1682 movies by 943 users. All ratings are integer values between one and five where one is the lowest (disliked) and five is the highest (most liked). Our data includes all the users who have rated at least 20 movies. We used a neighborhood size of 20 in the algorithms for both item-based and user-based techniques.

There has been considerable research in the area of recommender systems evaluation [4] .Our interest is along the lines of stability [6]: how the attack changes the system's ratings for the pushed item, but more generally we are interested in measuring the effectiveness of an attack - the "win" for the attacker. The desired outcome for the attacker in a "push" attack is of course that the pushed item be more likely to be recommended after the attack than before. In the experiments reported below, we follow the lead of [6] in measuring stability via prediction shift: the difference in the system's predicted rating for an item before and after the attack. A high prediction shift means that

[1] http://www.cs.umn.edu/research/GroupLens/data/

the attack has succeeded in making the system predict the attacked item as more preferred. To calculate an average value for prediction shift, we chose 50 movies at random from the MovieLens data, being careful that this set of target items mirrored the distribution of the data as a whole, and examine the impact of an attack against each movie over the whole user population.

Note that a strong prediction shift is not a guarantee that an item will be recommended. So, we also measure hit ratio, the average likelihood that a top N recommender will recommend the pushed item [7]. Over all trials, we count the number of times that the attacked item appears in recommendation sets of size N for different N. For our test set of movies, the pre-attack hit ratio is very small – less than 1% even for large recommendation set sizes. Post-attack hit ratio is therefore a good measure of the effectiveness of the attack on the pushed item compared to all other items.

For the segmented attack, we investigated two market segments: one defined by Harrison Ford's action movies and one by popular horror films. Recall that the segmented attack is constructed by identifying a set of segment items and the attacked users are those who have rated those items highly.[2] Due to space limitations, we report here only on the results from the Horror segment; the results in the other segment were similar. We chose those users who had given above average scores (4 or 5) to any three of the five movies. For this set of five movies, we then selected all combinations of three movies that had at least 50 users support, chose 50 of those users randomly and averaged the results.

For all the attacks, we generated a number of attack profiles and inserted them into the system database and then generated predictions. We measure "size of attack" as a percentage of the pre-attack user count. There are approximately 1000 users in the database, so an attack size of 1% corresponds to 10 attack profiles added to the system.

The segmented attack does not have a strong overall average impact. It is not as powerful as the average attack introduced by[5], for example. However, our assumption is that the attacker's primary interest is with the "in-segment" users, those users who have rated the segment movies highly and presumably are desirable customers for pushed items that are similar.

The intuition behind the segmented attack is borne out in Figure 1. The figure shows prediction shift results for the Horror segment, comparing all users with

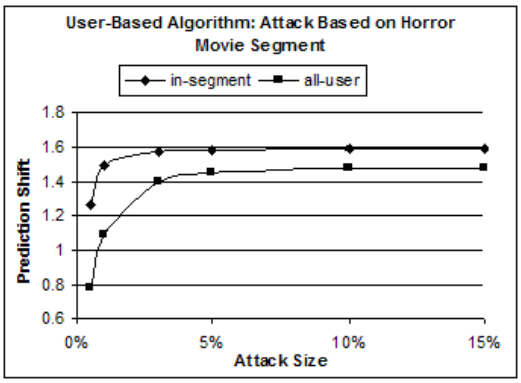

Figure 1. Prediction Shift results for the Horror Movie segment. User-Based algorithm.

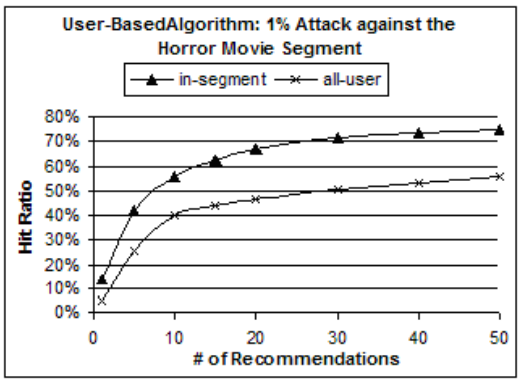

Figure 2. Hit Ratio results for the Horror Movie segment. User-Based algorithm.

in-segment users. The in-segment prediction shift is slightly stronger for the segmented attack than the average attack, the most effective attack we had previously studied.[3] The hit ratio results are shown in Figure 2 for a 1% attack at different values of N. These results show that even an attack as small as 1% on the user-based algorithm can have a significant impact on the hit ratio. It is also interesting that although the overall user base is not affected as much as the in-segment users, the shift is still significant.

The benefit of the segmented attack is considerably more striking in the item-based case shown in Figures 3 and 4. Lam and Reidl [5] concluded, based on their results with the random and average attacks, that item-

[2]In the Horror segment, the movies were *Alien*, *Psycho*, *The Shining*, *Jaws*, and *The Birds*. This list was generated from on-line sources of the popular horror films: http://www.imdb.com/chart/horror and http://www.filmsite.org/afi100thrillers1.html.

[3]Note also that the segmented attack requires considerably less knowledge of the ratings distribution in the system than the average attack requires. [1] discusses the question of limited knowledge attacks in greater detail.

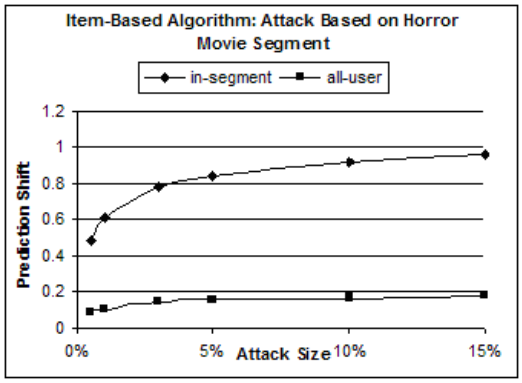

Figure 3. Prediction Shift results for the Horror Movie segment. Item-based algorithm.

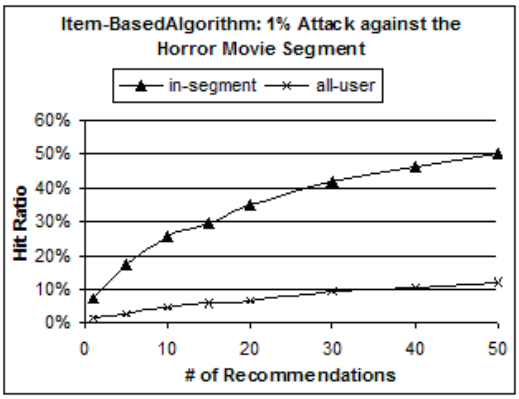

Figure 4. Hit Ratio results for the Horror Movie segment. Item-based algorithm.

based algorithms were more robust than user-based ones. However, as the figures show, the segmented attack works well against the item-based algorithm. Both of these figures show the focused manner in which this attack homes in on its target audience in the item-based algorithm. The general population is barely effected by the injected profiles, but there is a sizable prediction shift and hit ratio effect for in-segment users.

These results also point out an interesting difference between the user-based and item-based algorithms. While, is both cases, the attack has a dramatic impact on the in-segment users, the overall impact of the segmented attack on the whole user group is more pronounced in the case of user-based algorithm. Both the prediction shift and hit ratio results show that while the item-based algorithm remains vulnerable to this attack, it is more stable than the user-based algorithm.

5. Conclusions

Previous research has examined profile injection attacks against recommender systems that are broad in their construction and impact. Of these, the average attack has been found to be most effective. From a cost-benefit point of view, however, such attacks are somewhat wasteful: they require a significant degree of system-specific knowledge to mount; they push items to users who may not be likely purchasers and they are not effective against item-based implementations.

In this paper, we introduce the segmented attack, a profile injection attack that associates the pushed item with a small number of popular items of similar type. As our results show, the attack does well at ensuring the pushed item will be recommended to those users that are its target market. It is effective against item-based recommendation algorithms to a degree that broader attacks are not, and system-specific ratings distribution data is unnecessary.

References

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

[2] R. Burke, B. Mobasher, R. Zabicki, and R. Bhaumik. Identifying attack models for secure recommendation. In *Beyond Personalization: A Workshop on the Next Generation of Recommender Systems*, San Diego, California, January 2005.

[3] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, Berkeley, CA, August 1999.

[4] J.Herlocker, J. Konstan, L. G. Tervin, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004.

[5] S. Lam and J. Reidl. Shilling recommender systems for fun and profit. In *Proceedings of the 13th International WWW Conference*, New York, May 2004.

[6] M. O'Mahony, N. Hurley, N. Kushmerick, and G. Silvestre. Collaborative recommendation: A robustness analysis. *ACM Transactions on Internet Technology*, 4(4):344–377, 2004.

[7] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International World Wide Web Conference*, Hong Kong, May 2001.