Filler Item Strategies for Shilling Attacks against Recommender Systems

Sanjog Ray
Indian Institute of Management Calcutta, India
fp062004@iimcal.ac.in

Ambuj Mahanti
Indian Institute of Management Calcutta, India
am@iimcal.ac.in

Abstract

In recent years recommender systems have become a ubiquitous feature in e-commerce sites. However, the open nature of recommender systems makes them vulnerable to shilling attacks from malicious users. Such attacks may lead to erosion of user trust in the objectivity and accuracy of the system. One critical area of research in security of recommender systems is the study of attack models. In this paper, we propose an approach for creating attack models. Our paper explores the importance of target item and filler items in mounting effective shilling attacks. Our attack strategies are based on intelligent selection of filler items. Filler items are selected on the basis of the target item rating distribution. We propose filler item strategies for both all-user attacks and in-segment attacks. We show through experiments that our attack strategies are the most effective attack strategies against both user-based and item-based collaborative filtering systems.

1. Introduction

Recommender systems are technology-based systems that provide personalized recommendations to users. In these systems, opinions and actions of other users with similar tastes are used to generate recommendations. However, with increasing popularity of recommender systems in e-commerce sites, recommender systems have become susceptible to shilling attacks. In shilling attacks, attackers try to influence the system by inserting biased data into the system. Recent researches have started focusing on attack models and attack detection strategies. [1, 2, and 3]

An attack on a recommender system is mounted by injecting a set of biased attack profiles into the system. Each attack profile contains biased ratings data and a target item. Profiles are injected into the system by fictitious user identities created by the attacker. Every attack can be classified as a push attack or a nuke attack. In a push attack, the objective of the attacker is to increase the likelihood of the target item being recommended to a large section of the users in the system. While in a nuke attack, the objective is to prevent the target item from being recommended.

An attack is also classified as a *high-knowledge* attack or *low-knowledge* attack [3]. A *high-knowledge* attack requires more detail knowledge of the rating distributions of each item present in the system. While in a *low-knowledge* attack, to launch an attack, dependence on the recommender system for information on the items is minimal. The approach of constructing the attack profile, based on knowledge about the items, products, and users in a recommender system is known as an attack model. The primary objective of an attacker is to build attack models that provide the most impact with minimal knowledge. The other concern of importance to an attacker is to create models which are hard to detect by attack detection algorithms.

The general form of a push attack profile is shown in Figure 1 below. An attack profile consists of a set of m ratings for m items; where m is the total number of items present in the system. This attack profile of m ratings can be divided into four sets of items: a target item i_t, a set of selected items I_S, a set of filler items usually randomly chosen I_F, and a set of unrated items I_E. While mounting an attack, the set of selected items remains same for all the attack profiles. For example, if a set of 10 attack profiles are inserted into the system to mount a push attack then all the 10 attack profiles will have the same set of selected items, while the set of filler items for each profile will most likely differ as they are randomly selected. So, while mounting an attack, all attack profiles inserted into the system to construct the attack will have the same set of selected items and target item, but set of filler items may differ among the inserted profiles. Attack models are defined by the rules by which these four set of items are identified and the way ratings are assigned to the items present in the sets. For some attacks, set of selected items may be empty.

Selected Items (I_S)	Filler Items (I_F)	Unrated Items (I_E)	Target Items (i_t)

Figure 1. A General Form of Attack Profile

978-0-7695-3450-3/09 $25.00 © 2009 IEEE

Most research on attack models have mainly focused on the set of selected items while creating new attack models. In this paper, we discuss the importance of target item as a critical factor in constructing effective shilling attacks. We propose attack strategies based on target item rating distribution. Our attack strategies also examine the importance of filler items in improving the effectiveness of an attack. Unlike previous approaches, our attack strategies are tailored on the basis of the recommender system algorithm being used. Our paper provides different attack models for mounting effective attacks against user-based collaborative filtering systems and item-based collaborative filtering systems respectively. We also provide filler item strategies for effective attacks against in-segment users. This work is an approach that is focused on constructing attack strategies through intelligent use of filler items based on target item rating distribution. Through experimental evaluation we show that our proposed strategies can result in more effective attacks against both user-based and item-based recommender systems.

This paper is organized as follows. In section 2 we provide a brief summary of various attack models and their filler strategies. In section 3 we describe user-based and item-based collaborative filtering algorithm. The evaluation metric used is also described in section 3. A discussion on target item classification approach is presented in section 4. In section 5 and section 6 we provide details of our proposed filler items strategies for user-based and item-based collaborative filtering systems respectively. In section 7 we explain filler item strategies for segment-based attack. In section 8 we describe the experimental evaluation process and report the results obtained. We conclude the paper in section 9.

2. Types of attacks

Various attack models have been proposed in previous researches on shilling of recommender systems. We discuss below, some of the popular attack models on which much research is focused on. A comprehensive study of different attack models can be found in [3].

Random attack: One of the initial attack models, attack profile has filler items chosen randomly and ratings are assigned to the filler items from a set of random values chosen from a distribution centered on the system mean. System mean is the mean for all user ratings across all items. This is *a low – knowledge* attack, as minimal knowledge is required to obtain system mean value. It has been found that this model is not very effective [3].

Average attack: One of the most powerful attack models. In an average attack model, set of selected items is empty. Filler items are selected randomly, and each filler item is assigned its mean rating. Mean rating here corresponds to the average rating for the item across all users in the database who have rated it. Average attack is a *high-knowledge* attack, as mean rating of each filler item is required to mount an attack. However, in [4] it has been shown that this attack can be effective with limited knowledge i.e., an average attack model constructed using a small set of filler items can also result in an effective attack.

Bandwagon attack: In this model, the set of selected items contains few of those items that have high popularity among users. Thus, attack profiles constructed will result in creation of malicious users that will have higher chances of being similar to a large number of genuine users. Selected items and the target item are assigned maximum rating value. As in random attack, filler items are randomly selected and assigned mean rating of items across the whole system. Therefore, bandwagon attack can be seen as an extension of the random attack. Bandwagon attack is a *low–knowledge* attack, as popular items data can be obtained from publicly available information sources. But it has been observed that bandwagon attacks are not as effective as average attack [3].

Segmented attack: This attack is modeled to push the target item to those users who are most likely to be influenced by the recommendation. A segment is defined as a group of users having affinity for items of similar features. Group of users who have rated highly most of the popular horror movies is an example of a segment of users interested in horror movies. So, an attacker with intent to promote a horror movie will try to get his target item recommended to this segment of users, as the likelihood of influencing them is higher. In this model, the set of selected items contains few of those items that have high popularity among users of the targeted segment. Selected items and the target item are assigned maximum rating value. Filler items are identified randomly and given the lowest possible rating. It has been shown that segmented attack is the most effective model against in-segment users [3]. It is a *low-knowledge* attack, as selection of highly rated movie with similar features can be achieved from public information sources.

3. Recommendation algorithms and evaluation metrics

In this paper we have evaluated our filler item attack strategies against both user-based collaborative filtering algorithm and item-based collaborative filtering algorithm. In collaborative filtering (CF), a user is recommended items that people with similar tastes and preferences liked in the past. Unlike context-based systems that use subject keywords and demographic details to recommended items, CF technique mainly relies on explicit ratings given by the user and is the most successful and widely used technique [5]. In this section we describe the collaborative filtering algorithms, the evaluation metric used, and the notion of prediction shift.

3.1 User-based collaborative filtering

In user-based collaborative filtering [6], firstly, a neighborhood of *k* similar users is found for the target user. Then for generating prediction for an item not yet seen by the target user, weighted average of the ratings given by the *k* similar neighbors towards the predicted item is used.

To calculate similarity among users we use the Pearson-r correlation coefficient. Let the set of items rated by both users *u* and *v* be denoted by *I*, then similarity coefficient ($Sim_{u,v}$) between them is calculated as

$$Sim_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \overline{r}_u)(r_{v,i} - \overline{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \overline{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \overline{r}_v)^2}} \qquad (1)$$

Here $r_{u,i}$ denotes the rating of user *u* for item *i*, and $\overline{r}_u$ is the average rating given by user *u* calculated over all items rated by *u* .Similarly, $r_{v,i}$ denotes the rating of user *v* for item *i*, and $\overline{r}_v$ is the average rating given by user *v* calculated over all items rated by *v*.

To compute the prediction for an item *i* for target user *u,* we use the following formula.

$$P_{u,i} = \overline{r}_u + \frac{\sum_{v \in V} Sim_{u,v} (r_{v,i} - \overline{r})}{\sum_{v \in V} | Sim_{u,v} |} \qquad (2)$$

Where *V* represents the set of *k* similar users. While calculating prediction only those users in set *V* who have rated item *i* are considered.

3.2 Item-based collaborative filtering

In item-based collaborative filtering [7], similarities between the various items are computed. From the set of items rated by the target user, *k* items most similar to the target item are selected. For computing the prediction for the target item, weighted average is taken of the target user's ratings on the *k* similar items earlier selected. Weightage used is the similarity coefficient value between the target item and target user items.

To compute item-tem similarity we used adjusted cosine similarity. Let the set of users who rated both items *i* and *j* be denoted by *U*, then similarity coefficient ($Sim_{i,j}$) between them is calculated as

$$Sim_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \overline{r}_u)(r_{u,j} - \overline{r}_u)}{\sqrt{\sum_{u \in U} (r_{u,i} - \overline{r}_u)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \overline{r}_u)^2}} \qquad (3)$$

Here $r_{u,i}$ denotes the rating of user *u* for item *i*, and $\overline{r}_u$ is the average rating given by user *u* calculated over all items rated by *u.* Similarly, $r_{u,j}$ denotes the rating of user *u* for item *j*.

To compute the predicted rating for a target item *i* for target user *u,* we use the following formula.

$$P_{u,i} = \frac{\sum_{j \in I} Sim_{i,j} * r_{u,j}}{\sum_{j \in I} | Sim_{i,j} |} \qquad (4)$$

In equation 4, *I* represent the set of *k* most similar items to target item *i* that have already been rated by the target user *u*. As earlier mentioned, $r_{u,j}$ denotes the rating of user *u* for item *j*.

3.3 Prediction shift

For the purpose of measuring the effectiveness of the attack we use the widely used metric called prediction shift. Prediction shift of a target item is the difference of average predicted rate of the target item, after and before the attack, for all target users. Average Prediction shift of an attack is the average change in prediction for all target items. We use the same formula as in [3], which is defined as follows.

Let *U* and *I* be the sets of target users and target items. Let $\Delta_{u,i}$ denote the prediction shift for user *u* on item *i.* $\Delta_{u,i}$ can be measured as $\Delta_{u,i} = p'_{u,i} - p_{u,i}$, where $p'_{u,i}$ is the prediction value after the attack and $p_{u,i}$ before the attack. The average prediction shift for an item i over all users can be computed as

$$\Delta_i = \frac{\sum_{u \in U} \Delta_{u,i}}{|U|} \tag{5}$$

The prediction shift for an attack model is the average prediction shift for all items tested. It can be computed as

$$\Delta = \frac{\sum_{i \in I} \Delta_i}{|I|} \tag{6}$$

4. Target item classification

Our attack strategies are dependent on target item rating distribution. Target item is categorized into T_L or T_H category on the basis of the target item rating distribution. In a scale of 1 to 5, higher scale implies rating of 4 or 5, lower scale implies a rating of 1 or 2.

T_L: Target item with majority of their ratings at lower end of the rating scale fall into this category. In our experiment we grouped items with 60 % or more of their ratings as 1 or 2 in this category.

T_H: Target item with majority of their ratings at higher end of the rating scale fall into this category. In our experiment we grouped items with 60 % or more of their ratings as 4 or 5 in this category.

Once category of the target item and the recommendation system algorithm to be attacked are known, an appropriate strategy based on filler items is then used to construct attack profiles.

5. Filler item strategies for user-based collaborative filtering systems.

Known attack models like average attack, bandwagon attack, and segmented attack are focused at constructing attack profiles that will result in creation of malicious users that have greater chance of having high similarity with as many genuine users as possible. Because a malicious user with high similarity with a genuine user increases the chance of it being selected in top *k* similar neighbors of the genuine user, thereby influencing the rating of the target item. Bandwagon attack and segmented attack have tried to achieve this objective by selecting popular items as part of their attack profiles. An attack profile consisting of the popular items will have similarity with higher number of users, which should finally result in an effective attack. However, it has been found that average attack is more effective compared to bandwagon and segmented attack when effectiveness is measured across all users [3]. Segment attack performs better than average attack only in in-segment attacks. One possible reason for this unexpected result could be the presence of other factors which also affect the effectiveness of an attack.

Our proposed approach for user-based CF systems considers rating distribution of the target item as a critical factor that can affect the effectiveness of an attack. Unlike previous attack models which focus on creating malicious users that are similar to as many genuine users as possible from the set of all users; the objective of our approach is to create malicious users that increase similarity with as many of those genuine users who have rated the target item. Our approach proposes two strategies that are based on the rating distribution of the target item. Both strategies improve similarity by assigning appropriate values to filler items. As we show below, our proposed approach performs substantially better than best available attack model i.e., average attack model. We define below the two strategies.

5.1 Strategy UL

This strategy is followed when the target item falls in T_L category. As majority of the users have rated the target item at the lower end of the rating scale, to improve effectiveness of the attack, we need to create malicious users that are similar to the genuine users who have rated the target item with a lower value. So, to improve similarity, a randomly selected filler item is assigned the average rating given to the filler item by those users who have rated the target item at the lower scale.

Let U_A be the set of all users who have rated the randomly selected filler item I_F. Let U_L be the set of all users who have rated the target item at a lower scale of rating and have also rated the randomly selected filler item. So, in this strategy, filler item I_F is assigned the average rating given to it by the set of users $U_{L.}$ This approach differs from average attack in the way filler items are assigned values, as in an average attack item I_F was assigned the average rating given to it by the set of users $U_{A.}$

5.2 Strategy UH

This strategy is followed when the target item falls in T_H category. As a large majority of the users have rated the target item at the higher end of the rating scale, to improve effectiveness of the attack, we need to create malicious users that are similar to the genuine users who have rated the target item with a higher value. So, to improve similarity, a randomly selected filler item is assigned the average rating given to the filler item by those users who have rated the target item at the higher scale.

Let U_A be the set of all users who have rated the randomly selected filler item I_F. Let U_H be the set of all users who have rated the target item at a higher scale of rating and have also rated the randomly selected filler item. So, in this strategy, filler item I_F is assigned the average rating given to it by the set of users $U_{H.}$ This approach differs from average attack in the way filler items are assigned values, as in average attack item I_F was assigned the average rating given to it by the set of users $U_{A.}$

6. Filler item strategies for item-based collaborative filtering systems.

It has also been found that attack models are not as effective against item-based CF systems as they are against user-based CF systems [3]. One explanation of this may be because most attack models are designed to improve similarity among users than items.

Our proposed approach for item-based CF system is designed to improve similarity among items and considers rating distribution of the target item as a critical factor that can affect the effectiveness of an attack. Our approach proposes two strategies that are based on the rating distribution of the target item. Both strategies improve similarity by selecting appropriate filler items. While in filler item strategies against user-based CF systems we had modified an existing attack model i.e., average attack model by intelligently assigning values to filler items, the strategies elaborated in this section is a new approach that is focused on selection of appropriate filler items and is specifically built for attack against item-based collaborative filtering systems. As we show below, our proposed approach performs substantially better than average attack model. We define below the two strategies.

6.1 Strategy IL

This strategy is followed when the target item falls in T_L category. As majority of the users have rated the target item at the lower end of the rating scale, it is most likely that items similar to the target item will also be rated at the lower end of the scale. To improve effectiveness of the attack, we need to create malicious users that increase the similarity of the target item with items which are rated at the higher end of the rating scale. So, to improve similarity, we select filler items from the set of items which are highly rated by those users who have rated target item at a lower scale. In a scale of 1 to 5, higher scale implies rating of 4 or 5, lower scale implies a rating of 1 or 2.

Let U_L be the set of all users who have rated the target item at a lower scale of rating i.e., 1 or 2. Let I_F be the set of items that have been rated 4 or 5 by the set of users U_L. So, in this strategy, filler items are selected from this set of items I_F. To select filler items, frequency count of all items which are rated 5 in set I_F is computed and those with higher frequency count are given preference while selecting filler items. In case that the number of filler items required for creating an attack profile can't be fulfilled by all the 5 rated items in the set I_F then a frequency count of all 4 rated items in set I_F is computed. As in earlier case, during selection of filler items, items with higher count are given preference. It is also taken into consideration that filler items selected should be distinct. All items selected as filler items are assigned the maximum rating of the rating scale i.e., 5 in a scale of 1 to 5. In case number of filler items required are more than the number of items present in set I_F then remaining filler items are selected randomly from the set of all items as done in an average attack. In an average attack, filler items are randomly selected and rating assigned to a filler item is its average rating.

6.2 Strategy IH

This strategy is followed when the target item falls in T_H category. As majority of the users have rated the target item at the higher end of the rating scale, it is most likely that items similar to the target item will also be rated at the higher end of the scale. In this strategy we try to further increase the association of target item with more highly rated items. So, to improve similarity, we select filler items from the set of items which are highly rated by those users who have rated the target item at a higher scale. This will help further strengthen the similarity of target item with highly rated items.

Let U_H be the set of all users who have rated the target item at a higher scale of rating i.e., 4 or 5. Let I_F be the set of items that have been rated 4 or 5 by the set of users U_H. So, in this strategy, filler items are selected from this set of items I_F. The process of selecting filler items from the set I_F is exactly similar to that of Strategy IL.

The two strategies Strategy IL and Strategy IH differ in the way members of U_L and U_H are selected.

7. Filler item strategies for segment-based attack.

In [3] it has been shown that segmented attack is the most effective attack model against in-segment users. It performs much better than average attack model, the most effective model when effectiveness is measured across all users. A segmented attack is mounted with the intent of pushing a target item towards a specific group of users i.e., a segment of users that are grouped on the basis of their similarity in preferences for a category of items. The selection of segment is done by selecting highly rated popular items with similar characteristics. Segmented attack is focused on those users who have highly rated majority of the selected items present in the segment. For example, group of users who have highly rated at least any three of the five most popular animation movies form a segment of users interested in animation movies. An attacker with intent to promote a new animation movie will find it beneficial to mount a segmented attack against a segment of user interested in animation movies.

In our work, we study the effect of our filler item strategies in further improving the effectiveness of in-segment attacks against both user-based and item-based attacks. In segment attack, filler items are randomly selected and assigned the minimum value in the rating scale. The reasoning behind assignment of minimum value to the filler item has not been explained in detail in any of the literature on segmented attack. We provide below details of our filler item strategies for segment attack.

7.1 Strategy SUL

This strategy is followed when a segment attack is mounted against a user-based CF system and the target item falls in T_L category. In our approach to improve effectiveness of the attack, we need to create malicious users that are similar to those genuine users who have rated the target item with a lower value and also belong to the segment of users targeted by the segmented attack. So, to improve similarity, a randomly selected filler item is assigned the average rating given to it by those users who have rated any of the items that define the segment and have also rated the target item at a lower scale. For example, for an attack against a segment of users interested in animation movies i.e., those users who have rated highly at least any three of the five most popular movies in animation genre, a filler item is assigned the average rating given to it by those users who have rated the target item at a lower rating and have rated highly at least one of the five animation movies that define the segment.

7.2 Strategy SUH

This strategy is followed when a segment attack is mounted against a user-based CF system and the target item falls in T_H category. To improve effectiveness of a segment attack, a randomly selected filler item is assigned the average rating given to it by those users who have rated any of the items that define the segment and have also rated the target item at a higher scale.

7.3 Strategy SIL

This strategy is followed when a segment attack is mounted against an item-based CF system and the target item falls in T_L category. To improve effectiveness of a segment attack, we select filler items from the set of items which are highly rated by those users who have rated target item at a lower scale. The strategy used is similar to Strategy IL. Filler items are selected the way explained in section 6.1.

7.4 Strategy SIH

This strategy is followed when a segment attack is mounted against an item-based CF system and the target item falls in T_H category. To improve effectiveness of a segment attack, we select filler items from the set of items which are highly rated by those users who have rated target item at a higher scale. The strategy used is similar to Strategy IH. Filler items are selected the way explained in section 6.2.

8. Experimental evaluation and discussion

We performed the experimental evaluation of our strategies on the publicly available MovieLens data set [8]. This is the most widely used dataset in recommender systems research. MovieLens consists of 100,000 ratings made by 943 users on 1682 movies. Each user in the data set has rated at least 20 movies and each movie has been rated at least once. A timestamp value is associated with each user, movie, and rating combination. The data set also contains information on the demographic detail (age, sex, occupation, and zip code) of each user and basic information (genre and release date) of each movie. The ratings are made in a scale of 1 to 5, where 5 indicate extreme likeness for an item and 1 dislike.

We evaluated effectiveness of the proposed strategies on user-based and item-based collaborative algorithm. For similarity calculation and prediction in user-based CF algorithm, equations 1 and 2 stated in section 3 were used. Similarly, equations 3 and 4 stated in section 3 were used for computing similarity and prediction value for item-based CF algorithm. We used a neighborhood size of $k = 20$ for prediction calculation. Case amplification value of 10 was used while calculating correlation and only positive correlations values were considered for computing predictions.

To conduct our evaluation, we selected a sample 20 items. Out of the 20 items, 10 items belonged to T_L category while remaining 10 items to T_H category. All the 20 items were selected randomly from a larger set of items belonging to each category. We also randomly selected a sample of 50 target users. Target users selected were those who have never rated any of the 20 test items. Each of the target items was attacked individually and the prediction shift was calculated by averaging the prediction shift observed for each user. The final prediction shift for the attack is the average prediction over all items used in the test. Equation 6 was used to calculate the metric.

For implementation of segmented attack we followed the same guidelines as stated in [3]. Horror segment was selected as the target segment. Five of the most popular horror movies were selected to represent the segment. These five movies selected formed the selected item set in the attack profiles constructed. The five movies are *Alien, Psycho, The Shining, Jaws,* and *The Birds*. Users who have given a rating of 4 or 5 to at least any 3 of the five movies were identified as the target segment against which the attack was focused. For calculating prediction shift we selected 50 of the users from this target segment to form the test user set. While implementing the segment attack, selected items were given a rating of 5 and the randomly selected filler items were assigned a value 1.

All experiments were conducted for "S*ize of attack*" values 1%, 3%, 6%, 12%, and 15%. "S*ize of attack*" represents number of attack profiles added as a percentage of pre-attack profiles. 1% "S*ize of attack*" implies 10 attack profiles were added to a system of 1000 genuine users. On the basis of the results reported in [4] that best results are reported when a filler size of 3% is used in an average attack, we used a filler size of 3% for all our tests i.e., 3 % of 1682 items which is approximately 50 filler items. For attacks against user-based collaborative filtering systems we used six strategies: Strategy UL, Strategy UH, Strategy SUL, Strategy SUH, segment attack and average attack. Similarly, for attacks against item -based collaborative filtering systems we used six strategies: Strategy IL, Strategy IH Strategy SIL, Strategy SIH, segment attack and average attack. For average attack, filler item strategy used was the same as in an average attack i.e., the mean of the filler item was assigned to it. Segment attack was implemented as explained earlier. Category $T_{L,}$ Category $T_{H,}$ Strategy UL, Strategy UH, Strategy IL, Strategy IH Strategy SUL, Strategy SUH, Strategy SIL and Strategy SIH were implemented the way explained earlier in section 4, section 5, section 6 and section 7 respectively. For attacks against item-based CF while selecting filler items from set I_F, only items with minimum frequency count of 10 were considered.

Figure 2 and Figure 3 show the effectiveness of our attacks when calculated for all users against systems using user-based collaborative filtering for recommendations. Figure 2 shows the prediction shift values of attacks Strategy UL and average attack for items belonging to T_L category. From the graph it's obvious that for items in T_L category, Strategy UL outperforms average attack model for all values of attack size. Similarly, Figure 3 shows the prediction shift values for the attack strategies Strategy UH and average attack for items belonging to T_H category. From the graph it can be concluded that for items belonging to T_H category, Strategy UH performs much better than average attack over lower values of attack size. At attack size of 12 % and 15% both attack have similar effectiveness.

Figure 4 and Figure 5 show the effectiveness of our attacks when calculated for all users against systems using item-based collaborative filtering for recommendations. Figure 4 shows the prediction shift values of Strategy IL and average attack for items belonging to T_L category. Similarly, Figure 5 shows the prediction shift values for the Strategy IH and average attack for items belonging to T_H category. From Figure 4 and 5 it can be concluded that both Strategy IL and Strategy IH perform substantially better than average attack over all values of attack size. It can also be observed that our attack strategies are more effective against item-based systems than user-based systems.

Figure 6, 7, 8, and 9 show the effectiveness of our filler based attack strategies for in-segment users. We observe that for attacks against both user-based and item-based CF systems the effectiveness of our filler based strategies is comparable to the best available attack model for in-segment attacks i.e., segment attack. However, in Figure 8 we observe that filler item strategy SIL performs better than segment attack. Because of the low knowledge cost involved in segment attack, we can conclude that for most scenarios segment attack is a better attack model for in-segment attacks than filler based attack models.

Experimental results clearly show that our approach of selecting a strategy based on target item rating distribution outperforms the best available attack model i.e., average model. One drawback of our attack strategies is its high knowledge cost. However, automated software agents can help diminish the cost. One approach that can be used to decrease the cost is to use a subset of users while selecting filler items. For example, in attacks against item-based systems, while implementing Strategy IH instead of selecting all users who have rated target item as 4 or 5 as members of the set $U_{H.}$, we only select 20 users. Selection of items for set I_F will then be performed using the data of the 20 users in set U_H. Similarly, in case of attacks against user-based systems, while implementing Strategy UH instead of assigning a filler item I_F the average rating given to it by the set of users $U_{H.}$, we assign I_F the average rating given to it by a subset of 5 randomly selected users from $U_{H.}$ In future work we plan to experimentally verify the effectiveness of these cost reduction approaches.

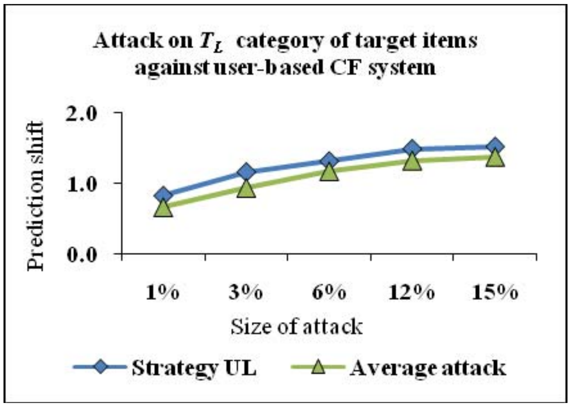

Figure 2: Attack on T_L category of items against user-based collaborative filtering system

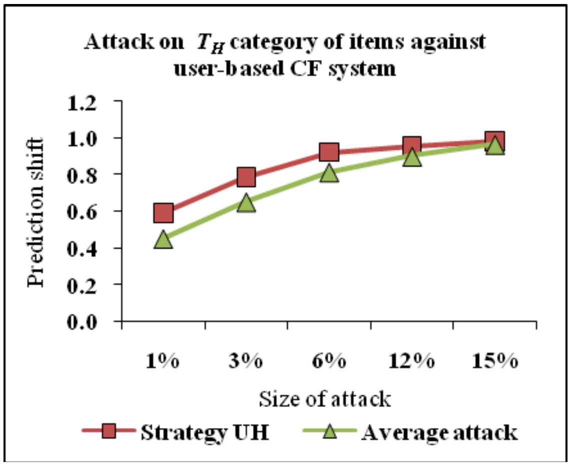

Figure 3: Attack on T_H category of items against user-based collaborative filtering system

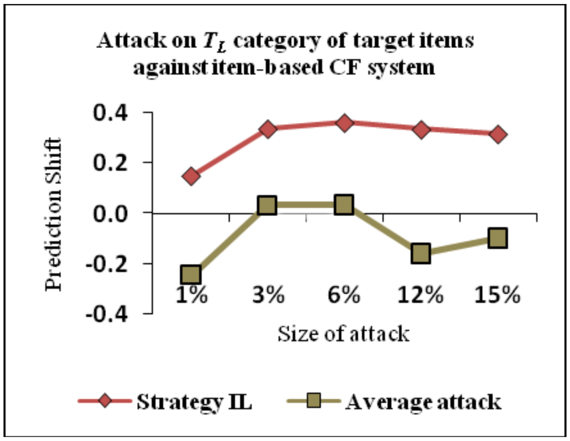

Figure 4: Attack on T_L category of items against item-based collaborative filtering system

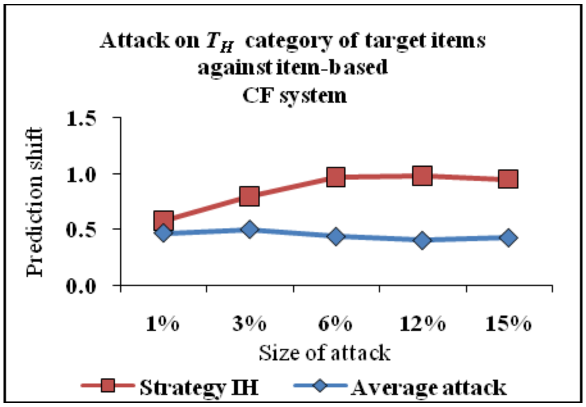

Figure 5: Attack on T_H category of items against item-based collaborative filtering system

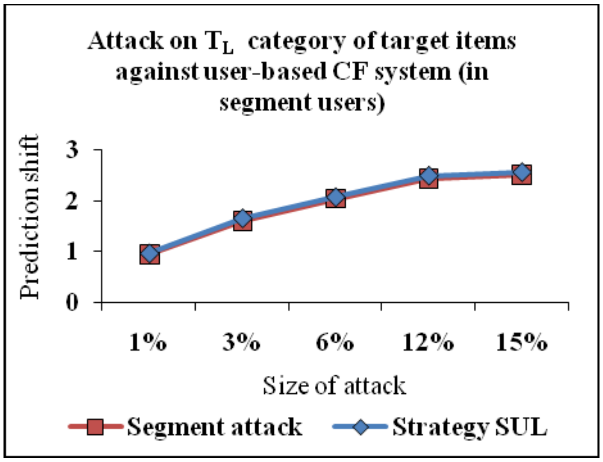

Figure 6: Attack on T_L category of items against user-based collaborative filtering system

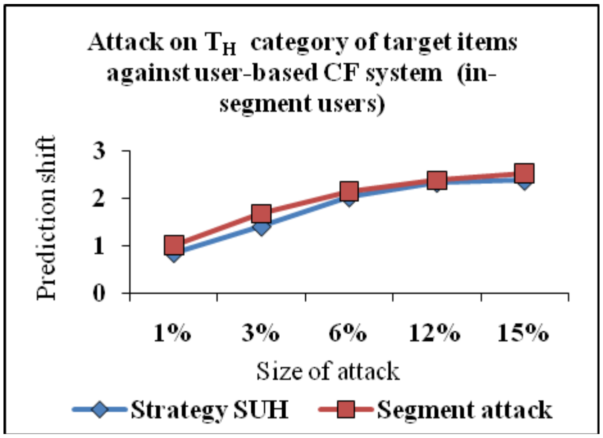

Figure 7: Attack on T_H category of items against user-based collaborative filtering system

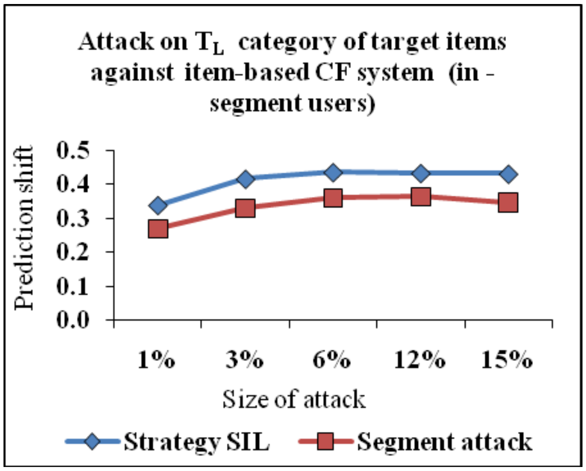

Figure 8: Attack on T_L category of items against item-based collaborative filtering system

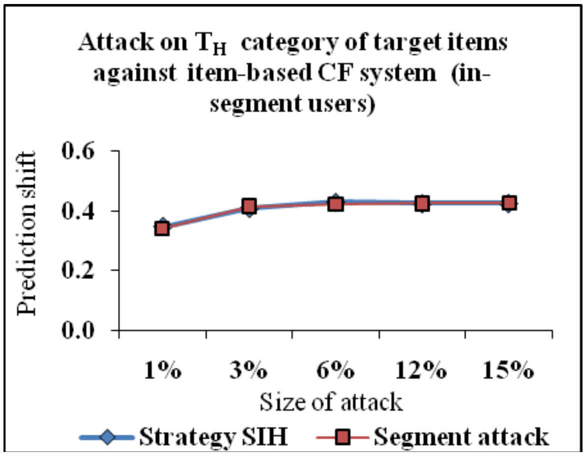

Figure 9: Attack on T_H category of items against item-based collaborative filtering system

9. Conclusion

This paper provides an effective approach towards constructing attack models. We show the importance of target item and filler items in construction of successful attack strategies. Through experiments we show that our approach of intelligent selection of filler items based on target item rating distribution results in substantial improvement over the baseline average attack. We also compare our approach with the well known in-segment approach and conclude that our approach gives slightly improved results. In future, we plan to examine the filler items strategies for other attack models, and also create algorithms to improve robustness and stability of recommender systems against shilling attacks.

10. References

[1] Lam, S., and Riedl, J. 2004. Shilling Recommender Systems for Fun and Profit, In Proceedings of the 13th International WWW Conference.

[2] Mehta, B., Hofmann, T., and Nejdl, W. 2007. Robust Collaborative Filtering, In Proceedings of the 2007 ACM Conference on Recommender Systems, 49-56.

[3] Mobasher, B., Burke, R., Bhaumik, R., and Williams, C. 2007. Towards Trustworthy Recommender Systems: An Analysis of Attack Models and Algorithm Robustness, ACM Transactions on Internet Technology, **7**(2007)**,** 23:1-38.

[4] Burke, R., Mobasher, B., and Bhaumik, R. 2005. Limited Knowledge Shilling Attacks in Collaborative Filtering Systems, In Proceedings of Workshop on Intelligent Techniques for Web Personalization.

[5] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J. 1997. GroupLens: Applying Collaborative Filtering to Usenet News, Communications of the ACM, 40, 3(1997), 77–87.

[6] Herlocker, J., Konstan, J., Borchers, A., and Riedl, J.1999. An Algorithm Framework for Performing Collaborative Filtering, In Proceedings of SIGIR, ACM, 77-87.

[7] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2001. Item-based Collaborative Filtering of Recommendation Algorithms. In Proceedings of the 10th International WWW Conference.

[8] MovieLens data set,www.grouplens.org