Attacking Item-Based Recommender Systems with Power Items

Carlos E. Seminario and David C. Wilson
Software and Information Systems Department
College of Computing and Informatics
University of North Carolina at Charlotte
Charlotte, NC 28223 USA
cseminar@uncc.edu davils@uncc.edu

ABSTRACT

Recommender Systems (RS) are vulnerable to attack by malicious users who intend to bias the recommendations for their own benefit. Research in this area has developed attack models, detection methods, and mitigation schemes to understand and protect against such attacks. For Collaborative Filtering RSs, model-based approaches such as item-based and matrix-factorization were found to be more robust to many types of attack. Advice in designing for system robustness has thus been to employ model-based approaches. Our recent work with the Power User Attack (PUA), however, determined that attackers disguised as *influential users* can successfully attack (from the attacker's viewpoint) SVD-based recommenders, as well as user-based. But item-based systems remained robust to the PUA. In this paper we investigate a new, complementary attack model, the Power Item Attack (PIA), that uses *influential items* to successfully attack RSs. We show that the PIA is able to impact not only user-based and SVD-based recommenders but also the heretofore highly robust item-based approach, using a novel multi-target attack vector.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval–*Information filtering*

Keywords

Recommender Systems; Power Item; Attacks; Evaluation

1. INTRODUCTION

Recommender systems help users decide which products and services to buy by making use of preference information provided by the community of system users. Unfortunately, these systems are subject to attack by malicious self-interested users who enter fake information to either promote their own products ("push"), disparage their competition ("nuke"), or to create havoc in the recommender [13, 8, 11]. Commercial system operators do not tend to disclose detailed information about attacks on their systems. But we know that real attacks on recommender systems are not uncommon, and they take on a variety of forms, the most popular of which is providing fake reviews known as opinion spam.[1] The problem with attacks on ratings-based Collaborative Filtering (CF) Recommender Systems (RS) is that predictions and recommendations can become biased in accordance with the type of attack perpetrated. In turn, attacks can potentially corrupt the system dataset and cause users to distrust the information and results provided by the system. In the absence of live attack data from service providers, research on RS attacks has focused primarily on similarity-based statistical models of user ratings, such as random and average hypothetical users [13, 8, 11]. Additionally, attack detection techniques have been developed based on these models of rating behavior, e.g., [11, 9, 18, 7]. Overall, model-based approaches such as item-based and matrix-factorization strategies were found to be more robust to many types of attack. The conventional advice in designing for system robustness has thus been to employ model-based approaches [12]. Because attackers continue to develop new approaches for biasing RS results, it is critically important for researchers, system designers and operators to keep pace in analyzing and understanding robustness characteristics and potential attack vectors.

We have previously studied a novel category of RS attacks based explicitly on measures of influence, in particular the potential impact of high-influence, or *power users* [19]. Power users in the RS context are those that are able to influence the largest group of RS users; influence is indicated by the ability of power user i to change (positively or negatively) the RS prediction for another user j, or for power user i's target item to appear in user j's top-N list. *To be clear, the power user attack in our research is not about having many actual power users collude to mount an attack, rather, it is about being able to generate a set of synthetic power user profiles that, when entered into a RS database, can effectively bias the recommendations.* We found that Power User Attacks (PUAs) are able to successfully impact SVD-based and user-based recommenders [16, 19, 20]; we also confirmed previous research [8, 11, 20] that item-based systems are fairly robust to attack.

In order to successfully attack (from the attacker's viewpoint) the item-based algorithm, we turned our attention to the complementary notion of influential *power items*. Selected in the same manner as power users, we conjectured whether power items would exhibit the same type of influence found with power users. Therefore, the overall question for this research is, **Can a Power Item Attack successfully impact item-based recommenders as measured with Hit Ratio, Prediction Shift, and Rank robustness metrics?** [2]

RecSys'14, October 6–10, 2014, Foster City, Silicon Valley, CA, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2668-1/14/10 ...$15.00.
http://dx.doi.org/10.1145/2645710.2645722.

[1] For example see, http://www.reuters.com/article/2013/09/23/us-fake-reviewers-idUSBRE98M0YU20130923

[2] See § 6 for description of robustness metrics.

This paper presents our definition of power items and the power item attack model, as well as a series of experiments conducted to determine how well the Power Item Attack (PIA) is able to impact the traditional item-based algorithm [15].

2. RELATED WORK

Attacking RSs by entering false ratings has been termed a *profile injection attack* [11] or *shilling attack* [8]. Burke et al provide a summary overview of RS attack models, attack detection, and algorithm robustness [2]. Most of the attack research has targeted the use of similarity-focused attack models that generate synthetic attack user profiles using random or average item ratings or a variant of these two approaches [8, 11]. Average attacks focus on attack coverage by broad similarity to all users or to a segment of RS users, and were shown to be more effective than Random attacks. Using these similarity-focused attack models, user-based systems were shown to be vulnerable to attack while item-based systems were more robust to this type of attack [8, 11]. Previous work on RS attacks has also indicated that recommenders using matrix factorization/SVD-based algorithms are robust to attack [9, 10]. However, this is only the case when attackers have been detected and removed from the recommendation process. None of these models used explicit measures of influence for coverage.

Our research has investigated using explicit measures of influence to create attack models based on the notion of *power users* [19]. We used well-known concepts from Social Network Analysis, i.e., Degree Centrality [17], and applied them to recommender systems. Furthermore, research showed that collaborative relationships in recommender systems can be represented as a social network [14]. In earlier work, we defined the *Power User Attack* model as a set of power user profiles with biased ratings that influence the results presented to other users [19]. The PUA consists of one or more user profiles containing item ratings (called attack user profiles) that push or nuke a specific item. The PUA demonstrated that influential users can impact recommendations for user-based and SVD-based systems; to a much lesser extent, item-based systems can also be impacted [19, 16, 20]. These attacks were successful because power users are able to correlate with many non-power users to impact the target item ratings.

Based on successful results with power users, we turn our attention to the complementary notion of *power items*, wherein attackers methodically select certain items to include in their attack user profiles that can be used to effectively influence other users and items in the RS. Power item identification has not been widely examined in the context of RS attacks, so our initial effort will use the same influence-based methods we used to select power users (see § 3). Certain attack models, such as the Bandwagon, Segment, and Average over Popular (AOP) attacks [1, 11, 7], specified the use of popular items (those with many ratings) in the attack user profiles to correlate with selected groups of users. While these attacks were successful (from the attacker's viewpoint) against user-based algorithms, only the Segment attack was found to be effective against a small subset of users using an item-based recommender. However, none of these attack models were successful against the full complement of users in the dataset. The item-based algorithm has proven to be robust against these attacks and this remains an open challenge in RS robustness research that we explore in this study.

3. SELECTING POWER ITEMS

To select *power items* our initial study employs the same methods we used previously for power user selection [20]. We believe this is sound for similarity-based methods because the similarity calculations between items are symmetric to those between users. The methods are as follows:

InDegree or ID: Our approach is based on in-degree centrality [17], where power items participate in the highest number of similarity neighborhoods. For each item i compute similarity with every item j applying significance weighting $n_{cij}/50$, where n_{cij} is the number of users that have rated the same items i and j, then discard all but the top-N neighbors for each item i.[3] Count the number of similarity scores for each item j (# neighborhoods item j is in), and select the top-N item j's.

Aggregated Similarity (AggSim or AS): Analogous to the user-based Most Central heuristic from [4]. The top-N items with the highest aggregate similarity scores become the selected set of power items. This method requires at least 5 users who have rated the same item i and item j; this method does not use significance weighting.[4]

Number of Ratings (NumRatings or NR): Power users were defined in [6] as users with the highest number of item ratings, thus the analog for power items would be those items with the highest number of user ratings. Therefore, we select the top-N items based on the total number of user ratings they have in their profile. Items selected by this method are also referred to as popular items in the context of Bandwagon, Segment, and AOP attacks [1, 11, 7].

When evaluating power item selection methods, there are attack dimensions such as cost and knowledge required that should be considered [8, 2]. The cost to mount an attack is controllable by the attacker and relates to the effort required to yield the desired outcome; the objective is to keep the cost low. The more knowledge an attacker has about the dataset's users, items, and ratings, the more effective the attack; however, that knowledge is difficult, albeit not impossible, to obtain. We note here that the knowledge required for the NumRatings method can be considerably lower than InDegree or AggSim because popular items are usually well known and are publicly-available information; this may give NumRatings an edge over the other selection methods, costs being equal.

Although PIA detection is beyond the scope of this paper, we should note that detailing the Power Item Model (§ 4) and the methods for selecting power items (§ 3) provides the basic information required for detection analysis.

4. POWER ITEM MODEL

We have developed a Power Item Model (PIM) that can be used to generate synthetic power item profiles (SPIP) for attack purposes. Unlike classic attack models (e.g., random, average, bandwagon) that employ straightforward statistical templates (e.g., average item rating, popularity, and likability) to generate synthetic attack profile filler items [11], very little is known about the characteristics of power items. And without this knowledge, it is difficult to build attack user profiles. So, for the PIA, our initial work uses influence-based methods to select power items (§ 3) and we set other attack user profile elements in the SPIP according to more traditional attack models.

To describe the PIM, we use the specification framework from [11]. The attack user profile elements consist of the following:

Selected items (I_S) have particular characteristics determined by the attacker. For the PIM, these are the power items and they are items that are likely to correlate with many user profiles in the system. The selected item size, or the number of items in each profile, is an experimental design parameter and is usually expressed as a percentage of the total number of items in the dataset. A larger size

[3] We used a divisor of 50 users as an analog to work done with co-rated items in user neighborhoods by [5] to optimize RS accuracy.

[4] Based on personal communication with the authors.

may have more impact, however, it is also more easily detectable. Previous work [11] has shown that a 5-10% profile size should be sufficient to have an impact on recommendation robustness. I_S selection is based on the methods described in § 3. The I_S rating value for each of these items in the profile is selected randomly from a normal distribution around the mean and standard deviation of the item's rating in the dataset. Our intent was for SPIP's to have a rating profile that was strong rather than just randomly assigning rating values. We used a normal distribution because this has been typical in RS attack research. [8, 11].

Filler items (I_F) are usually set randomly according a normal distribution and are used to establish correlations with other users in the dataset. For the PIM, this set is empty because we wanted a strong correlation between the selected items I_S and the target item I_T; we believe that having filler items would tend to confound or dilute this relationship.

Unrated items (I_U) are the items exclusive of the I_S, I_F, and I_T and have null values in the PIM.

Target item (I_T) is usually a single item that is typically set to the maximum r_{max} or minimum r_{min} rating depending on the attack intent (push or nuke). Our initial experiment in this study consisted of a *single target item* attack (PIA-ST) in keeping with traditional attack models; our subsequent experiments (2 and 3) used the novel *multiple target item* attack (PIA-MT) on the item-based algorithm. The selection of the target item is also a key part of the attack model. We experiment with "new" items (those with only one rating) because this is a typical scenario in which power users are asked to provide ratings and because items with few ratings are more vulnerable to attack; we also use a mix of "new and established" items for subsequent experiments.

Other factors in building effective RS attacks include [8, 11]:

Attack size, the number of attack user profiles to be injected. A larger attack size may be more effective, however, it is more easily detectable. The attack size or number of profiles is an experimental design parameter and is usually expressed as a percentage of the total number of user profiles in the dataset. Previous work [11] has shown that a 5-10% attack size should be sufficient to have an impact on recommendation robustness. We vary the attack size for our experiments to understand the scope of impact.

Attack intent, for a typical 1-5 rating scale, 5 is used for push attacks and 1 for nuke attacks. In this study, we focus on push attacks, leaving nuke attacks for future work.

Therefore, to generate a set of SPIP's for a given PIA, we specify the following elements:

- Dataset with (user, item, rating) triples
- Power Item selection methods: similarity-based, influence-based, etc.
- Attack size or number of attackers: Expressed as a percentage of number of users in the dataset
- Selected Item (I_S) size or number of power items: Expressed as a percentage of number of items in the dataset
- Target Item (I_T): New items, Established items
- Target Item size or number of target items: Expressed as a percentage of number of items in the dataset
- Attack intent: Push

The push version of the PIA-ST is similar to the Bandwagon, Segment, and AOP attacks, when the power item selection method is based on the Number of Ratings method, as described in § 3. However, these attack models differ primarily in the contents of I_F and I_S as shown in Table 1. Furthermore, the PIA-MT differs radically from previously studied attacks using popular items, not only in the profile contents shown in Table 1 but also in that the PIA-MT uses multiple targets simultaneously rather than a single target item in order to mount the attack.

Table 1: Attack Model Profile Content Differences

Attack Model	I_S	I_F
Bandwagon	Popular items, ratings set to r_{max}	Random items, ratings set with normal dist around *system* mean
Segment	Segment items, ratings set to r_{max}	Random items, ratings set to r_{min}
Average Over Popular	Empty	x-% Popular Items, ratings set with normal dist around *item* mean
Power Item	Power items, ratings set with normal dist around *item* mean	Empty

The PIM approach goes beyond prior research primarily in two areas: first, we utilize influence-based methods (§ 3) to select the power items for the attack user profile, and second, we utilize multiple rather than just single target items. We believe that this combination can yield powerful attacks, especially against the item-based algorithm that has been resistant to attack in the past [8, 11].

5. ANALYZING POWER ITEM ATTACKS

We conducted a series of three experiments to address our main research question — whether the PIA could have a substantial impact on item-based recommenders. First, to see whether the PIA had traction as an attack vector overall, which it did. Second, to see whether a multiple-target variant would have a greater impact on item-based approaches, which it did. And third, to see whether the multiple-target PIA could have an impact on both new and established items, which it can. The line of experimentation was to find a PIA approach that was more successful in attacking item-based recommenders than previous research [11] had indicated.

Experiment 1: Consists of the PIA with a number of "new" (low # ratings) item targets pushed one at a time and averaged over all target items. We call this the PIA Single Target (PIA-ST) attack because we are, in effect, attacking the recommender with a single target item. The objective of this experiment is to determine the effectiveness of the PIA against various recommender algorithms and to compare with the results we obtained with the PUA against similar recommenders.

Experiment 2: Although it is easy to envision an attacker with an intent to promote a single item, e.g., a book they just published, it is also possible for an attacker to have several items to attack at once in order to promote (or disparage) a group of products as opposed to only one product. This experiment consists of the PIA with multiple "new item" targets all pushed at the same time and is called the PIA Multiple Target (PIA-MT). The objective of this experiment is to test how well the power item approach can significantly impact item-based systems, above and beyond previously-observed results by further exploiting item-item similarities in the SPIP's.

Experiment 3: A question that also needs to be answered is whether the PIA can still be effective when using a mix of new and established target items rather than just new items. This experiment consists of the PIA with multiple "new and established item" targets all pushed at the same time and is another variation of the PIA-MT. The objective of this experiment is to determine how well the PIA-MT is able to impact recommendations for a mix of new and established items.

Based on our research question, we note two hypotheses:

H1: *A PIA with relatively small number of SPIP's (<=5% of all users) can have significant effects on RS predictions and top-N lists of recommendations, measured with robustness metrics.* For Experiments 1 and 2 that use new items as targets, we expect Hit Ratio to be $>$ 50% and Rank $<$ 20 to qualify as significant impacts. For Hit Ratio, a majority of users ($>$ 50%) should have target items in their top-N lists. In our experiments we use a top-N value of 40 for Hit Ratio calculations based on the analysis in [8] that the median recommendation search ends within the first 40 items displayed. Therefore, a Rank of 20 would be well within the median search. Since there is no precedent for measuring a PIA that uses new and established items as targets, for Experiment 3 we used values based on the "all-users" Hit Ratio and Prediction Shift results for the Segment attack against the item-based systems [1, 11], i.e, Hit Ratio $>$ 11% and Prediction Shift $>$ 0.1.
H2: *SPIP's identified using the InDegree power user selection method will have a higher level of impact, compared to SPIP's identified using NumRatings or AggSim, on RS predictions and top-N recommendation lists as measured with Hit Ratio and Rank.* This hypothesis is based on the findings from Social Network Analysis [17] that high InDegree centrality is indicative of nodes (users) that have strong influence over other users.

6. EXPERIMENTAL DESIGN

Evaluation Metrics: Evaluations were performed before and after the attacks using the Apache Mahout 0.8 platform[5]. For robustness metrics [11, 2], we use Hit Ratio (HR), Average HR ($\overline{HR}$), Prediction Shift (PS), Average PS ($\overline{PS}$), Rank (R), and Average R ($\overline{R}$). For example, a high Hit Ratio and a low Rank indicates that the attack was successful (from the attacker's standpoint). Since we are using multiple targets simultaneously in Experiments 2 and 3, the interpretation of Hit Ratio is changed from its traditional meaning, i.e., HR is now the percentage of users that have at least one of the multiple target items in their top-N list. We also defined a new metric, Number of Targets per User (NTPU), associated with Hit Ratio that provides the average number of target items present in a user's top-N list of recommendations. This metric provides a measure of the effectiveness of a multiple-item attack, a higher NTPU meaning higher attack effectiveness, and is averaged over all users with hits (target items in their top-N lists). For a test run T, let U_T be the set of users, UH_T the set of users with hits, and IT_T the set of target items; and let R_u be the set of top-N recommendations for user u. If the target item appears in R_u for user u, the scoring function H_{ui} has value 1; otherwise it is zero. NTPU for a user u is given by $NTPU_u = \sum_{i\epsilon IT_T} H_{ui}$, and then averaged over all users with hits to yield $NTPU = \frac{\sum_{i\epsilon U_T} NTPU_u}{|UH_T|}$. To compare the NTPU metrics within and between experiments, a normalized NTPU or NNTPU is calculated using average Hit Ratio as the normalizing factor. So, for a given test run T, $NNTPU_T = \overline{HR_T} * NTPU_T$. Since the PIA's being evaluated for Experiments 1 and 2 are for "new" items, i.e., items with one rating, the Prediction Shift is expected to be close to r_{max} of 5.
Datasets and Algorithms: We used MovieLens[6] ML100K[7], ML1M[8], and ML10M[9] datasets. The RS algorithms used were provided in Apache Mahout and customized for this study. The CF user-based weighted algorithm (UBW) [3] uses Pearson similarity with a threshold of 0.0 (positive correlation), neighborhood size of 50, and significance weighting of n/50 where n is the number of co-rated items [5]. The item-based weighted algorithm (IBW) [15] uses Adjusted Cosine similarity with a threshold of 0.0 and significance weighting of n/50. For the SVD-based algorithm (SVD), we used RatingStochasticGradientDescent (RSGD); run-time parameter settings were number of features (=100) and number of training steps or iterations (=50) and were determined empirically to optimize recommender accuracy.

Table 2: Attack Parameters by Dataset

	Attackers	**Power Items**	**Target Items**
MovieLens 100K:			
% of Dataset	1, 5	1, 5, 10	1, 5
# Attackers, Items	10, 50	17, 83, 166	10, 50
MovieLens 1M:			
% of Dataset	0.1, 1	0.1, 1, 10	0.5, 1, 3
# Attackers, Items	6, 60	4, 37, 368	18, 37, 110
MovieLens 10M:			
% of Dataset	0.1, 1	0.1, 1, 10	0.5, 1
# Attackers, Items	70, 699	11,107,1068	50, 100

Attack User Profiles: To mount the Power Item Attack, attack user profiles were generated as described in § 4 and converted to attack profiles by setting target items to the Attack Intent.
Power Item Selection: Methods used for power item selection are described in § 3.
Target Item Selection: For Experiments 1 and 2, we used 'new' items, i.e., target items with only one rating were selected randomly from the corresponding dataset. Experiment 3 used "new and established" items, i.e., target items were selected randomly and had the following average number of ratings, average rating, and average rating entropy, respectively: ML100K (73.78, 3.13, 1.77), ML1M (253.40, 3.26, 1.81), Ml10M (675.76, 3.15, 1.71).
Attack Parameter Selection: The Attack Intent is Push, i.e., target item rating is set to max (= 5). The Attack Size or number of power users in each attack was varied for these experiments; the I_S size (number of power items) and the number of target items used were also varied as shown in Table 2. The Attack profiles were generated as described in § 4 and the target item rating was injected at run time.
Test Variations: For all three experiments, we used all three power item selection methods (§ 3). For Experiment 1 we used UBW, IBW, and SVD algorithms, ML100K and ML1M datasets, and single new target items. For Experiments 2 and 3 we focused on the IBW algorithm and used all three datasets. Experiment 2 used new multiple target items and Experiment 3 used new and established multiple target items.

7. EXPERIMENTS AND RESULTS

7.1 E1: PIA-ST with "New" Item Targets

Single target item attacks have been used in the past [11, 2] to eliminate confounds between the selected/filler items (that are used to correlate with other users) and the target item. This is especially important for user-based recommenders because user-user similarities with multiple target items would form neighborhoods of users that have similar tastes not only with selected/filler items but also

[5] http://mahout.apache.org

[6] http://www.grouplens.org

[7] nominal 100,000 ratings, 1,682 movies, and 943 users.

[8] nominal 1,000,209 ratings, 3,883 movies, 6,040 users.

[9] nominal 10,000,054 ratings, 10,676 movies, 69,878 users.

with the multiple target items, effectively reducing the focus of the attack. To successfully attack item-based systems, prior research showed that item-item similarities can be manipulated; e.g., this was demonstrated in the Bandwagon and Segment attacks [1, 11].

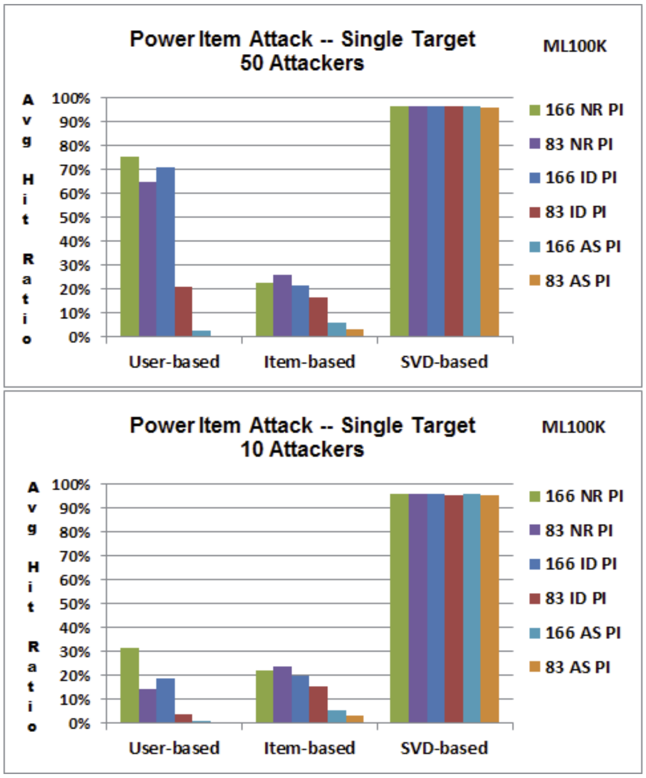

Figure 1: ML100K – Experiment 1 Hit Ratio Results

For this experiment we select 50 target items from the ML100K dataset that only have one rating, with the intent of attacking a "new" item. We calculate impacts on robustness metrics (see § 6) for each target item individually and then average the results over all 50 targets. We repeat this calculation for three levels of power items (17, 83, and 166) and two levels of SPIP's (10 and 50) for each of the three power item selection methods (InDegree, NumRatings, and AggSim) and using each of the three recommender algorithms (UBW, IBW, SVD). The Hit Ratio results for ML100K are shown in Figure 1. For the case with 50 attackers or 5% of user base (top of Figure 1), both InDegree and NumRatings show strong $\overline{HR}$ results using 166 power items for UBW and SVD (70% to 75% for UBW and 96% for SVD) and significantly weaker results for IBW (21% to 22%). AggSim shows strong results for SVD (96%) and very weak results for UBW and IBW ($< 5\%$). Results for $\overline{R}$ (not shown) indicate little variation across the power item selection methods and average as follows: 3.0 for UBW, 14.6 for IBW, and 2.0 for SVD. And results for $\overline{PS}$ (not shown) also indicate little variation across the power item selection methods and are at a higher level (> 4) because of the "new" item targets. For the case with 10 attackers or 1% of user base (bottom of Figure 1), we observe similar results against IBW and SVD as well as a significantly weaker attack against UBW.

To see whether these results would scale, we repeated a similar experiment using the ML1M dataset for two levels of power items (37 and 368) and two levels of SPIP's (6 and 60) for each of the three power item selection methods (InDegree, NumRatings, and AggSim) and using each of the three recommender algorithms (UBW, IBW, SVD). Results for this ML1M attack (not shown) using 60 attackers (1% of user base) and each with 368 power items (10% of all items), are similar to those obtained for ML100K with 10 attackers (also 1% of user base), i.e., a weak attack for UBW (high of 21% to 36% $\overline{HR}$) and IBW (13% to 19% $\overline{HR}$), and a strong attack for SVD (81% to 98% $\overline{HR}$). This would indicate that more attackers are required for a stronger attack. Results for $\overline{R}$ average as follows over all power item selection methods: 6.7 for UBW, 15.4 for IBW, and 5.5 for SVD.

Overall, these results indicate that under a specific set of conditions (e.g., using 50 attackers and 166 power items for ML100K), the PIA is effective (high $\overline{HR}$, low $\overline{R}$) against the UBW and SVD algorithms. We also found that the PIA is not very effective against the IBW, regardless of the test conditions. While we had hoped to see a larger impact on IBW using the PIA, our results are consistent with previous findings (including our PUA) [8, 11, 19], showing that the item-based algorithm is resistant or robust to attack. Hypothesis H1 is accepted for both UBW and SVD recommenders, meaning that a relatively small number of power users (5% or less of the user base on a given dataset) can have significant effects on RS predictions and top-N lists of recommendations regardless of power user selection method. IBW is partially accepted because $\overline{R}$ < 20, however, $\overline{HR}$ does not meet the 50% requirement. Hypothesis H2 is rejected for all three algorithms. Although the InDegree and NumRatings perform well at a high level, NumRatings is a slightly better method for selecting power items, i.e., simply inserting popular items into SPIP's creates very effective attacks against some recommender systems (UBW and SVD in our experiment).

7.2 E2: PIA-MT with "New" Item Targets

The motivation for Experiment 2 was to develop a PIA model that had higher impacts on IBW than had been previously observed. Intuitively, we expect for carefully configured single-item attacks such as Average, Bandwagon, and Segment attacks [8, 11, 2] to be effective against user-based algorithms because of the similarity correlations established between the selected and filler items of the attacker profiles and the corresponding items in the profiles of non-attackers in the dataset. Once that strong correlation is made (by the algorithm), then the correlation between the selected/filler items and the target item allows the algorithm to calculate a higher prediction value for the target item which is then recommended to the non-attacker. Previous results indicate that larger attack and filler sizes create stronger attacks and research has shown that these attack models consistently impact user-based systems with impunity [8, 11, 2]. The item-based algorithm, however, establishes similarity correlations between the selected/filler items and the target item of the attacker profiles that are then used to calculate recommendations for non-attackers. The Segment attack [11, 1] was successful against the item-based algorithm to the extent that it impacted users who belonged to a particular segment of the user base (e.g., the "Horror" movie crowd), however, this attack did not have a high impact over the entire user base. We believe that to mount a stronger attack against item-based systems, two elements are required in the attack user profile. First, the set of selected items must correlate with a broad cross-section of the user base and second, multiple target items must be used to establish strong correlations with the selected items. Experiment 2 takes on this challenge.

We recognize that because of multiple target items, there can be impacts to the robustness metrics, i.e., the $\overline{HR}$ for a single target item will be different due to confounding when a target item is grouped with multiple other target items during the similarity and prediction calculation process. An analysis of this situation was performed and we found that a metric such as Hit Ratio decreases slightly for any given target item as the number of multiple target items in the SPIP increase. For example, for a set of attacks using ML100K and IBW, we found that the HR for a single target item across all users decreased from 0.225 to 0.208 to 0.184 going from 1 to 10 to 50 targets, respectively. However, at the same time, the $\overline{HR}$ across all target items and users increased from 22% to 70%, so the confounding effect for IBW does not present a major issue for the PIA.

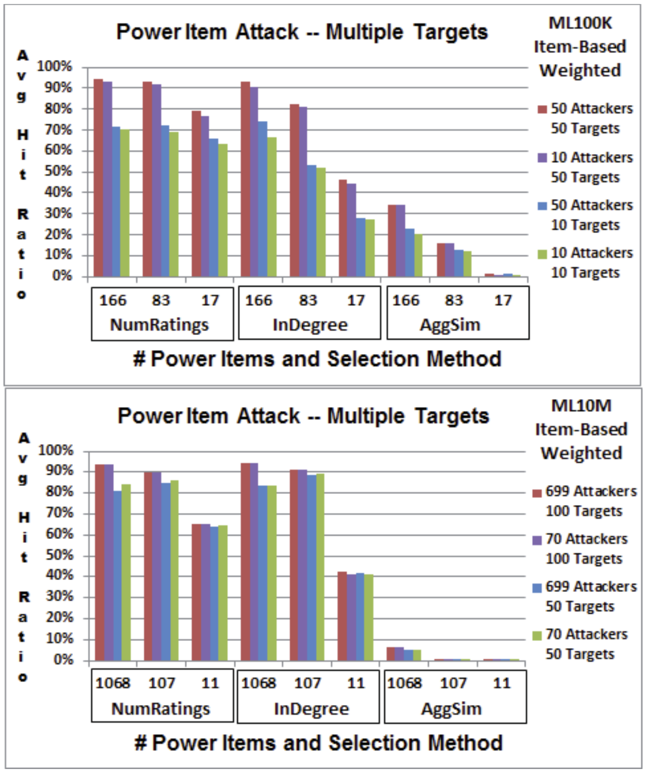

Figure 2: ML100K / ML10M – Experiment 2 Hit Ratio Results

The effectiveness of Experiment 2 was measured using robustness metrics (see § 6). For each dataset used in this experiment, we select a specified number of target items that only had one rating with the intent of attacking "new" items. Those target items are injected into the dataset at one time and then $\overline{HR}$, $\overline{R}$, and $\overline{PS}$ impacts over all targets are calculated. This process is repeated for three levels of power items, up to three levels of SPIP's for each of the three power item selection methods (InDegree, NumRatings, and AggSim), and using only the IBW recommender algorithm. See Table 2 for parameter settings. The ML100K $\overline{HR}$ results shown in the upper chart of Figure 2 indicate some interesting characteristics for this type of attack. InDegree and NumRatings show strong $\overline{HR}$ values (80% to 90%) when attack profiles used 166 and 83 power items and 50 target items, while AggSim impacts were weaker (15% to 34%) for the same number of power items and target items. Average Hit Ratio is sensitive to the number of power items and target items, and somewhat insensitive to number of attackers; *i.e., the PIA can be effective with a small number of attack user profiles*. NumRatings shows the least amount of this sensitivity across the number of power items and target items, i.e., 10 attackers, each with 17 power items and 10 target items (a total of 270 ratings) impacts over 60% of the user base with 100,000 ratings.

To see if these results scaled, we also ran this experiment on ML1M and ML10M. For the ML1M dataset (not shown), we observed a similar set of characteristics in the results. InDegree and NumRatings continue to show strong $\overline{HR}$ results while AggSim results are much weaker. And a NumRatings attack with 6 attackers, each with 4 power items and 37 target items (a total of 888 ratings) impacts 40% of the user base with a million ratings. For the ML10M dataset, results are shown in the lower chart of Figure 2. InDegree and NumRatings continue to show strong $\overline{HR}$ results while AggSim results are much weaker. And a NumRatings attack with 70 attackers, each with 11 power items and 50 target items (a total of 38,500 ratings or 0.4% of the total number of ratings) impacts 64% of the user base with ten million ratings. Average Rank for each of these cases was also calculated: for ML100K, $\overline{R}$ varied from 9 to 19 (mean 15.9); for ML1M, $\overline{R}$ varied from 14 to 19 (mean 16); and for ML10M, $\overline{R}$ varied from 10 to 20 (mean 16). To compare the attack effectiveness within and between datasets in the experiment, the NNTPU metric is shown in Figure 3. The highest number of targets per user occurs with the SPIP's containing 11 power items and 100 target items generated using the NumRatings method for ML10M; a close second would be SPIP's containing 107 power items generated using the InDegree method. For all three datasets, the results for $\overline{PS}$ (not shown) indicate little variation across the power item selection methods and are at a higher level (> 4) because of the "new" item targets. To further confirm our results, we also ran a complete set of baseline PIAs across all datasets, attack sizes, and power item levels for IBW *without any target items*. The robustness metrics were all zero, meaning that injecting SPIP's without any target item ratings had no effect on the RS recommendations.

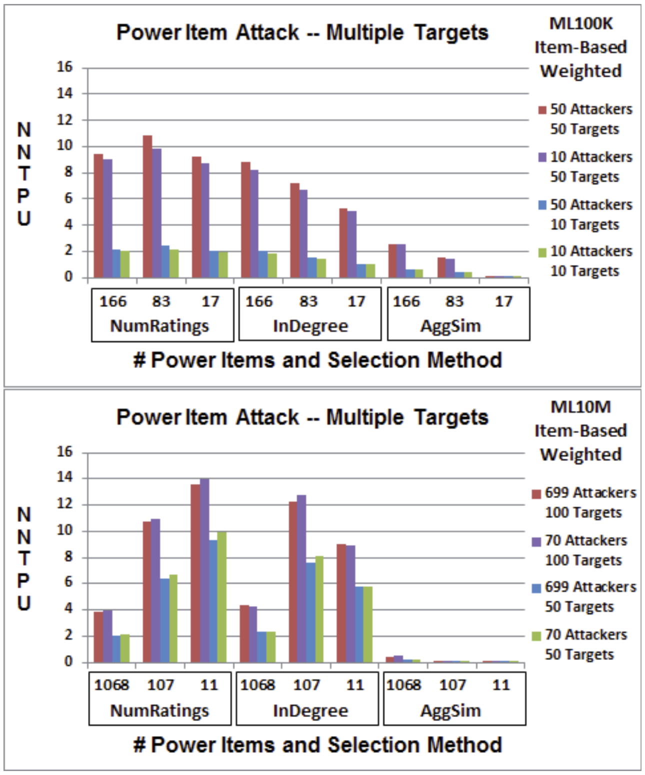

Figure 3: ML100K / ML10M – Experiment 2 Normalized Number of Targets Per User (NNTPU) Results

Most notable is that the $\overline{HR}$ results exceed Hit Ratio measurements reported previously for attacks against item-based recommenders, including the Segment attack. We conclude that the use of power items and multiple (new) target items in the SPIPs has resulted in a powerful attack against the item-based algorithm. Hypothesis H1 is accepted for the higher levels of attack size and number of targets for all power item selection methods and all three datasets. Hypothesis H2 is partially accepted for the IBW algorithm. Although the InDegree and NumRatings both perform well at a high level, NumRatings is a slightly better method for selecting power items, especially at lower levels of power items; both methods are superior to AggSim.

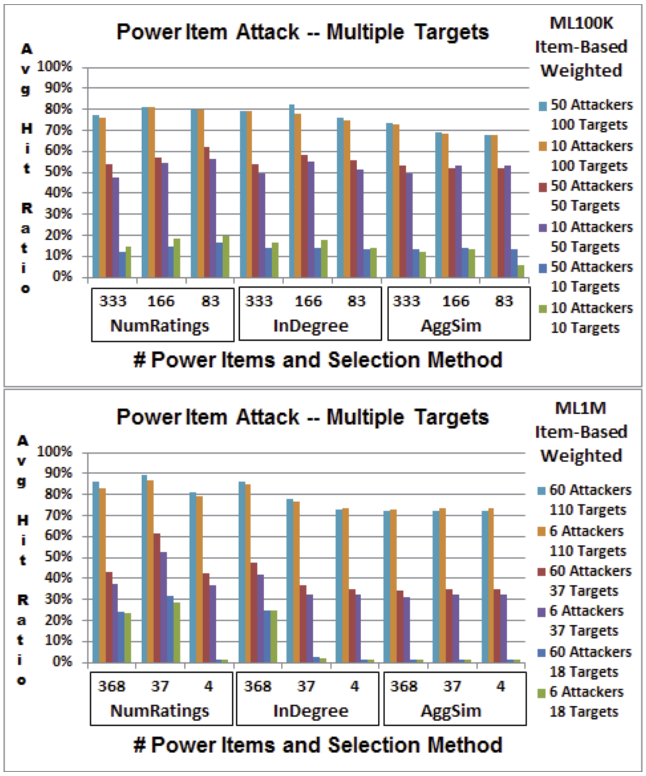

Figure 4: ML100K / ML1M – Experiment 3 Hit Ratio Results

7.3 E3: PIA-MT with "New and Established" Item Targets

In general, the robustness results for this experiment were lower than Experiment 2; this was expected since new items are more vulnerable to attack than established items. For each dataset used in this experiment, we select a specified number of target items to obtain a mix of items with a range of "age" based on number of ratings. The attack and calculation processes described for Experiment 2 are used again here. See Table 2 for parameter settings. For the ML100K dataset, we added a third level of SPIP's with 100 target items (10% of the total number of items) to compare with the three levels of target items in ML1M and to observe the impacts resulting from adding more target items to the SPIP's. The $\overline{HR}$ results shown in the upper chart of Figure 4 indicate sensitivity to the number of target items and insensitivity to number of attackers and power items. A similar pattern can be observed for the ML1M dataset shown in the lower chart of Figure 4; this is also the case for ML10M (not shown) except for the sensitivity to the number of power items for NumRatings and InDegree. For higher numbers of target items, ML100K and ML1M show strong $\overline{HR}$ results across all power item selection methods; for ML10M, NumRatings and InDegree still have a slight edge (40% to 50%) over AggSim (31%) although not quite as substantial as in Experiments 1 and 2. Average Rank for each of these cases was also calculated: for ML100K, $\overline{R}$ varied from 17 to 21 (mean 19.6); for ML1M, 18 to 23 (mean 20.6); and for ML10M, 19 to 21 (mean 20.3).

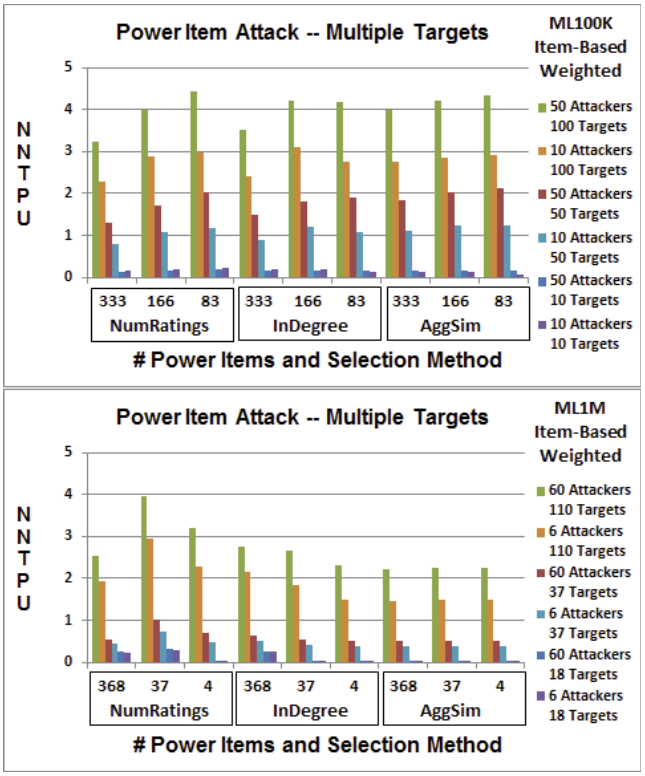

Figure 5: ML100K and ML1M – Experiment 3 Normalized Number of Targets Per User (NNTPU) Results

To compare the attack effectiveness between Experiments 2 and 3, we used the NNTPU metric shown in Figure 5 for ML100K and ML1M. The results confirm that attacks in Experiment 2 had more impact than those in Experiment 3. For example, for ML100K and 50 target items, Experiment 2 had NNTPU values between 4.5 and 11 for NumRatings and InDegree, AggSim had values between 0 and 2. Experiment 3 had NNTPU values between 1.3 and 2.1 for all three selection methods. An interesting result for ML100K is that NNTPU displays a phenomenon similar to one reported in previous work, i.e., as the number of power items increases, the attack effectiveness decreases (see upper chart in Figure 5); this occurs consistently for all three power item selection methods. Reported in [11], as the number of filler items increases, PS decreases; the explanation for this was that attack user profiles need to achieve a balance between "coverage" (including enough item ratings to correlate with other users) and "generality" (including too many item ratings that could make the profile dissimilar to a given user). We also observed this for NumRatings and InDegree for ML10M in this experiment (not shown) and in Experiment 2 (see Figure 3).

Regarding Prediction Shift results, for ML100K we observed $\overline{PS}$ values in the range of 0.2 to 0.4 and for ML1M they ranged from 0.03 to 0.19. By comparison, [1, 11] reported PS values of 0.1 and 0.15 for the Segment and Bandwagon attacks, respectively, against the item-based algorithm for all users in ML100K with an attack size of 1%. Our $\overline{HR}$ and $\overline{PS}$ results for ML100K were significantly improved over previously reported results. Given time constraints, full re-implementation, testing, and execution of Segment / Bandwagon attacks for more direct comparison was beyond the scope of the experiment. It is a limitation of the study to be addressed in future work.

To further determine the quality of our results, we computed $\overline{PS}$ for attack datasets that included the power items but not the target items and compared results statistically. For ML100K and 100 target items, differences in $\overline{PS}$ with and without the target items were significant (p <0.005) for NumRatings, InDegree, and AggSim across all three levels of power items. For ML1M, differences were significant (p <0.05) for NumRatings and InDegree (368 power items only). As in Experiment 2, we ran a set of PIA's across all datasets, attack sizes, and power item levels for IBW *without any target items* as a baseline. In this case, the robustness metrics were all $>$ zero. The interpretation is that, because Experiment 3 uses "new and established" items as target items, it is possible (and expected) that some of them will show up in top-N recommendation lists as confirmed by our findings. However, we found significant differences in key metrics for cases with and without targets. For example, averaged over all the cases run with ML1M, NNTPU was 2.29 (with targets) and 1.38 (without targets) and $\overline{PS}$ was 0.09 and 0.01, respectively; this indicates that the attack had impacts above and beyond the baseline.

Hypothesis H1 is accepted for the highest levels of attack size and number of targets across all power item selection methods for ML100K and ML1M, given the threshold rates of 11% $\overline{HR}$ and 0.1 $\overline{PS}$. H1 is partially accepted for ML10M for $\overline{HR}$. Hypothesis H2 is partially accepted for the IBW algorithm. We find that the InDegree and NumRatings methods, on average, perform the same at all levels of power items and both methods are superior to AggSim.

8. CONCLUSION

In this study we have developed a power item model that is able to generate synthetic power item profiles that can be used to mount effective power item attacks against user-based and SVD-based recommenders measured by traditional Hit Ratio, Rank, and Prediction Shift robustness metrics. In addition, we showed how the power item attack using a novel multi-target approach can generate effective attacks against the typically robust item-based algorithm using new, as well as established, dataset items. We have also compared power item selection methods used to generate synthetic power item profiles and shown that, because of its low-cost and low-knowledge requirements, the NumRatings method is the more effective, by a small margin, in attacking recommenders than the influence-based InDegree method. We have shown that a relatively small number of NumRatings and InDegree synthetic power item profiles can have significant effects on RS predictions and top-N recommendation lists. And, in order to compare attack effectiveness results within and between our experiments, we developed a metric that measures the number of target items per user resulting from the multi-target approach. Future work includes evaluation in other domains / datasets and developing PIA detection methods.

9. REFERENCES

[1] R. Burke, B. Mobasher, R. Bhaumik, and C. Williams. Segment-based injection attacks against collaborative filtering recommender systems. In *Proceedings of the International Conference on Data Mining*, 2005.

[2] R. Burke, M. P. O'Mahony, and N. J. Hurley. Robust collaborative recommendation. In F. Ricci et al., editors, *Recommender Systems Handbook*. Springer, 2011.

[3] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendations methods. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*. Springer, 2011.

[4] A. Goyal and L. V. S. Lakshmanan. Recmax: Exploiting recommender systems for fun and profit. In *Proceedings KDD*, 2012.

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

[6] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 2004.

[7] N. Hurley, Z. Cheng, and M. Zhang. Statistical attack detection. In *Proceedings of the third ACM conference on Recommender systems*, 2009.

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

[9] B. Mehta and W. Nejdl. Attack resistant collaborative filtering. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2008.

[10] B. Mehta and W. Nejdl. Unsupervised strategies for shilling detection and robust collaborative filtering. *User Modeling and User-Adapted Interaction*, 19(1-2), 2009.

[11] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Trans. Internet Technol.*, 2007.

[12] B. Mobasher, R. Burke, and J. Sandvig. Model-based collaborative filtering as a defense against profile injection attacks. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'06)*, July 2006.

[13] M. P. O'Mahony, N. Hurley, and G. C. M. Silvestre. Promoting recommendations: An attack on collaborative filtering. In *Proceedings of DEXA'02*, 2002.

[14] J. Palau, M. Montaner, B. Lopez, and J. L. D. L. Rosa. Collaboration analysis in recommender systems using social networks. In *Cooperative Information Agents VIII: 8th International Workshop, CIA 2004*, 2004.

[15] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the World Wide Web Conference*, 2001.

[16] C. E. Seminario and D. C. Wilson. Assessing impacts of a power user attack on a matrix factorization collaborative recommender system. In *Proceedings of the 27th Florida Artificial Intelligence Research Society Conf.*, 2014.

[17] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, New York, NY, 1994.

[18] C. Williams, B. Mobasher, and R. D. Burke. Defending recommender systems: detection of profile injection attacks. *Service Oriented Computing and Applications*, 2007.

[19] D. C. Wilson and C. E. Seminario. When power users attack: assessing impacts in collaborative recommender systems. In *Proceedings of the 7th ACM conference on Recommender Systems*, RecSys '13. ACM, 2013.

[20] D. C. Wilson and C. E. Seminario. Evil twins: Modeling power users in attacks on recommender systems. In *Proceedings of the 22nd Conference on User Modelling, Adaptation and Personalization*, 2014.